EXPERT SYSTEMS APPLICATIONS

for the Electric Power Industry

Volume 2

Joseph A. Naser

Electric Power Research Institute

EXPERT SYSTEMS APPLICATIONS FOR THE ELECTRIC POWER INDUSTRY

EXPERT SYSTEMS APPLICATIONS FOR THE ELECTRIC POWER INDUSTRY

VOLUME 2

Edited by

Joseph A. Naser
Electric Power Research Institute
Palo Alto, California

Sponsored by

ELECTRIC POWER RESEARCH INSTITUTE
Palo Alto, California

HEMISPHERE PUBLISHING CORPORATION
A member of the Taylor & Francis Group

New York Washington Philadelphia London

EXPERT SYSTEMS APPLICATIONS FOR THE ELECTRIC POWER INDUSTRY

1 2 3 4 5 6 7 8 9 0 EBEB 9 8 7 6 5 4 3 2 1 0

Cover design by Renée E. Winfield.
A CIP catalog record for this book is available from the British Library.

Library of Congress Cataloging in Publication Data

Expert systems applications for the electric power industry / edited
by Joseph Naser : sponsored by Electric Power Research Institute.
p. cm.
Includes bibliographical references and index.

1. Electric power-plants—Data processing—Congresses. 2. Expert systems (Computer science)—Industrial applications—Congresses. 3. Nuclear power plants—Data processing—Congresses. I. Naser, Joseph. II. Electric Power Research Institute.
TK1191.E974 1990
621.31′0285—dc20 90-4717
CIP

ISBN 1-56032-102-4
EPRI: NP-6957

Contents

TECHNOLOGY, TOOLS, AND METHODS

NUCLEAR POWER PLANT APPLICATIONS

FOSSIL POWER PLANT APPLICATIONS

NUCLEAR PERFORMANCE APPLICATIONS

APPLICATIONS

METHODOLOGIES

TUTORIAL SESSION

Preface

The Nuclear Power, Generation and Storage, and Electrical Systems Divisions of the Electric Power Research Institute (EPRI) sponsored the Conference on Expert System Applications for the Electric Power Industry, which was held in Orlando, Florida, on June 5-8, 1989. The conference was hosted by Florida Power Corporation and Duke Power Company. It was attended by a diverse group of over 300 representatives of electric utilities, equipment manufacturers, engineering consulting organizations, universities, national laboratories, and government agencies. It consisted of a keynote address, 90 papers, 5 tutorial presentations and 3 luncheon presentations by authors from 13 countries. In addition, 25 application systems were demonstrated in the evenings. EPRI has performed and sponsored a substantial effort in advancing the field of expert systems for the electric power industry. Thirty-three papers and 12 demonstrations presented at this conference discussed EPRI-related activities.

Experts from 15 countries were brought together to discuss expert systems applications in the electric power industry. The results of a survey at the end of the conference showed that attendees were impressed with the wide variety of applications that exist or are being developed for the electric power industry. The conference described many expert systems that have already been tested and implemented or are currently in an advanced stage of development. This focus on production grade systems may be contrasted to a meeting just two years ago, when most applications were in the planning or early developmental stages. Thus, this conference marks a major step forward in expert system technology for the electric power industry.

The purpose of this technology transfer conference was to stimulate vigorous efforts to deploy expert system technology by increasing a large and diverse awareness of the number and variety of expert system applications available to the electric power industry. The participants left the conference with a sense of excitement that expert system applications have matured enough to offer immediate and substantial benefits for the electric power industry in a wide variety of domains, including operations, maintenance, and planning. These benefits include increased

productivity and efficiency, improved quality, enhanced safety, improved consistency and objectivity, reduced costs, and finally, improved methods for capturing, packaging, and distributing corporate expertise.

Joseph Naser

SESSIONS

POSTER PRESENTATIONS

An Expert System Assisting Geothermal Reservoir Characterization

J. ARELLANO, V. M. ARELLANO, and E. IGLESIAS
Instituto de Investigaciones Electricas
Apartado Postal 475
Cuernavaca, Morelos, 6200, Mexico

ABSTRACT

This paper presents ANAPPRES V2.0, the second version of a computerized expert system that interprets data of constant and variable-flowrate interference tests in which there are an arbitrary number of production wells and an arbitrary number of observation wells. From the analysis, the transmissivity (kh/), storativity (ch), and hydrologic boundaries (existence, type and location) are obtained. ANAPPRES successfully couples a mathematical model, an optimization technique and heuristic knowledge, an unusual combination rarely found in published expert systems. The main advantages of ANAPPRES as compared to standard methods, are: (1) it can analyze variable flowrate interference tests; (2) it can obtain 4 to 5 parameters (transmissivity, storativity, type of boundary, distance and angle to boundary) in one computer run; (3) it is considerably faster than a human expert in completion of the job and (4) allows analysis of well tests including an arbitrary number of observation wells and an arbitary number of production wells.

INTRODUCTION

Exploited geothermal fields are large-scale heat recovery systems [1]. These systems are composed of a heat source (often a shallow magmatic intrusion), a geothermal reservoir, a variable number of wells drilled into the reservoir, and appropiate surface facilities. The reservoir consists of a porous-permeable rock formation, and a fluid (brine, steam, non-condensable gas). Heat is transferred from the source to the reservoir, mined through wells via fluid extraction, and brought to its final use by the surface facilities. Final uses include power generation and space and process heating.

The deliverability of the fluid, and therefore that of the mined heat, is highly dependent upon the details of the local hydraulic transmissivity [(permeability*reservoir thickness)/fluid viscosity]. Furthermore, in-situ fluid reserves depend strongly on local storativity [porosity*composite fluid-rock compressibility*reservoir thickness] and reservoir area. Therefore, there are powerful incentives to assess transmissivity, storativity and reservoir area. These quantities are usually determined by means of pressure transient tests performed in wells.

The general method of transient pressure tests [2, 3] is to subject the reservoir to a known perturbation, record the pressure history response, and interpret it in terms of previously developed mathematical models. Well test analysis always requires matching the observed pressure histories to model-predicted histories. Matching provides transmissivity and storativity values, as well as other parameters of interest. There are two general types of pressure transient tests: those in which only one well is involved, and those in which more than one well is involved. Perturbations are affected via so called active wells; pressure history recording can take place both in active wells, and in so called observation wells.

Interference tests, in which one or more active wells, and one or more observation wells participate, have distinct advantages over conventional well tests in which only one well is involved. Interference tests usually produce [3] transmissivity and storativity values averaged over greater areas, can detect anisotropies in those properties, and can provide information about the existence of hydrologic boundaries, their type (e.g., no-flow, constant-pressure) and their location.

For interference test analysis, a type-curve graphical technique ([e.g., 3]) has been the standard of the trade for many years. In practice, this technique has the disadvantages of being restricted to relatively simple cases and of requiring subjective judgement for curve fitting. The popularization of digital computers brought about computerized analysis techniques (e.g., [4]). These techniques, by use of regression and least-squares linear programming, eliminated the subjectivity previously associated with fitting a model to the observations, and provided the possibility of studying complicated systems and handling large quantities of data. However, the application of these techniques still requires extensive experience on the part of the analyst, and, except in the simpler cases, is laborious. The labour is associated with the necessity of running the same program many times, in cleverly selected sequences, with different subsets of data. The experience is required mainly for providing initial guesses of the parameter solutions to start the programs, for applying adequate quantitative criteria for accepting a computed fit to the data and for selecting the proper sequence in which to run the program and the corresponding data subsets.

This work describes ANAPPRES V2.0 (ANAlizador de Pruebas de PRESión, Spanish for "Well Test Analyst"), a computerized expert system developed to analyze interference tests in homogeneous reservoirs. ANAPPRES is user friendly, requires essentially no experience on the part of the analyst, eliminates subjectivity, can handle complex cases and large data sets, and completes the analysis of even the most complex cases, including plotting the results, in one run. In the current version ANAPPRES can analyze interference tests including an arbitrary number of active wells and an arbitrary number of observation wells. The user can be a beginner: the only requirement is that he (she) can create computerized files with the test data, for input. If prompted by the user, ANAPPRES explains how and why it arrived at the current conclusion. This feature has didactic advantages for non-expert users, provides verification capabilities for expert analysts and increases confidence in the

expert system. These characteristics of ANAPPRES were obtained applying artificial intelligence techniques, mathematical models, optimization techniques, heuristic knowledge and graphics software.

ANAPPRES was developed on a VAX-11/780 computer system. For input-output it uses the terminal Digital VT340, and for hardcopy output the Digital LA210 letterprinter and the Hewlett Packard HP-7585 pen plotter. ANAPPRES is written in FORTRAN 77.

ARCHITECTURE OF ANAPPRES

The architecture of ANAPPRES is presented in figure 1. There are 5 main modules, with 4 of them (the user interface, the computational module, the knowledge base and the explanatory module) linked to the central inference engine which drives the analysis. The functions of the different modules are described below.

User Interface

Its main goal is to provide a friendly environment for communication with the user. The main functions of this module are to generate menus, to display diagnostics and numerical results, to generate graphics, and to display explanations. Figures 2-5 illustrate the presentation of menus, results, graphics and explanations, respectively.

Computational Module

This is a modified version of the program ANALYZE [4]. The modifications to the original code include refreshing the memory in succesive calls to this module, automatic handling of error conditions, and the inclusion of criteria to decide whether a run with given reservoir parameter inicial guesses will converge.

ANALYZE was developed for analysis of interference tests in single (liquid) phase, homogeneous, isotropic reservoirs. This program determines reservoir parameters by minimizing the differences between observed and calculated pressure histories. Pressure histories are calculated with the Theis solution. The minimization is achieved by means of a non-linear least-squares routine. A Chi-square statistic, normalized to the observed pressures, provides a quantitative measure of the goodness of the fit.

Knowledge Base

The knowledge base is organized in production rules, with the well-known IF-THEN format [5]. It contains the knowledge necessary to perform the analysis of interference tests. This knowledge includes quantitative criteria to decide whether there is a hydrologic boundary and its type, what initial guesses to use in order to start the computational module, etc. For example figure 6 is a map in the transmissivity-storativity plane, illustrating 16 trigger points, and the corresponding areas of convergence, that ANAPPRES will use as initial guesses to start the computational module, if the user chooses not to provide an initial guess, or if

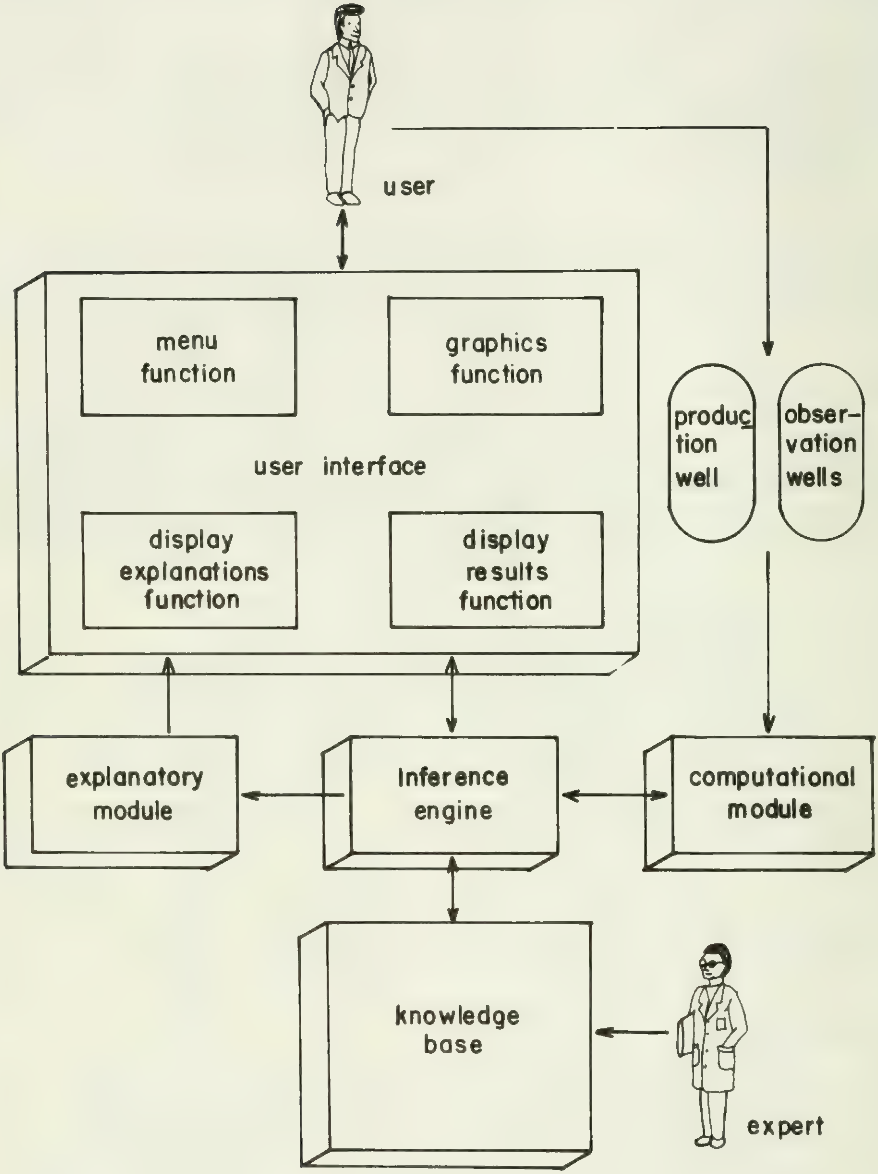


Figure. 1. Architecture of ANAPPRES.

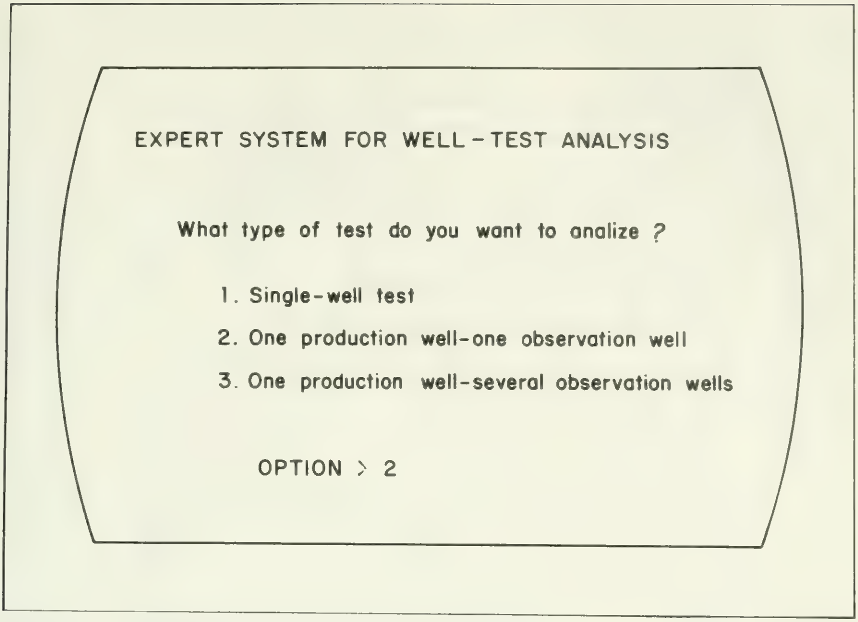


Figure. 2. Example of a display generated by the menu function.

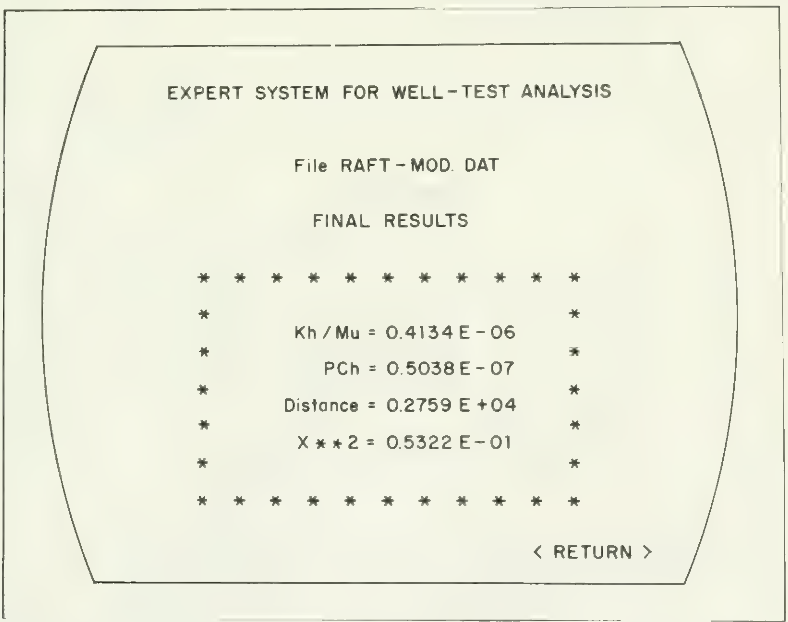


Figure. 3. Example of a screen generated by the display results function.

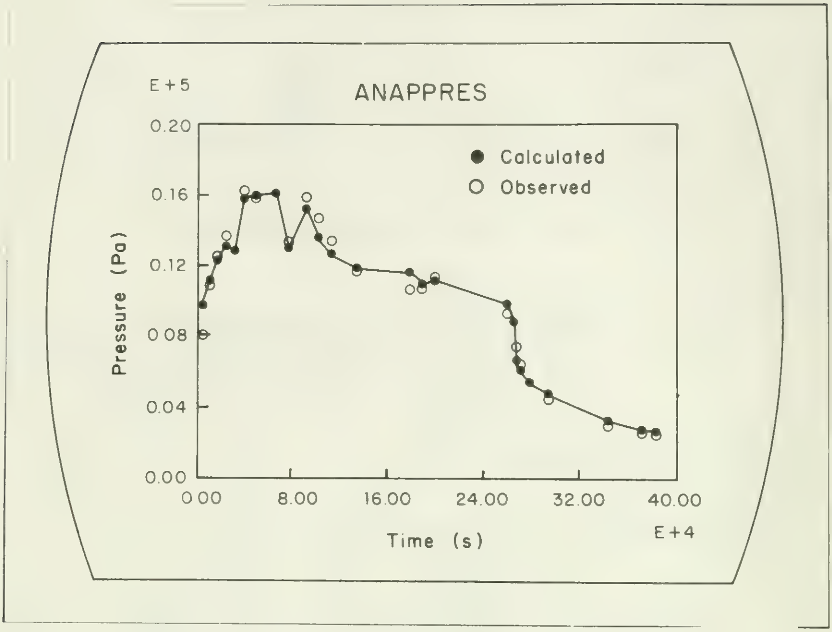


Figure. 4. Example of a screen generated by the graphics function.

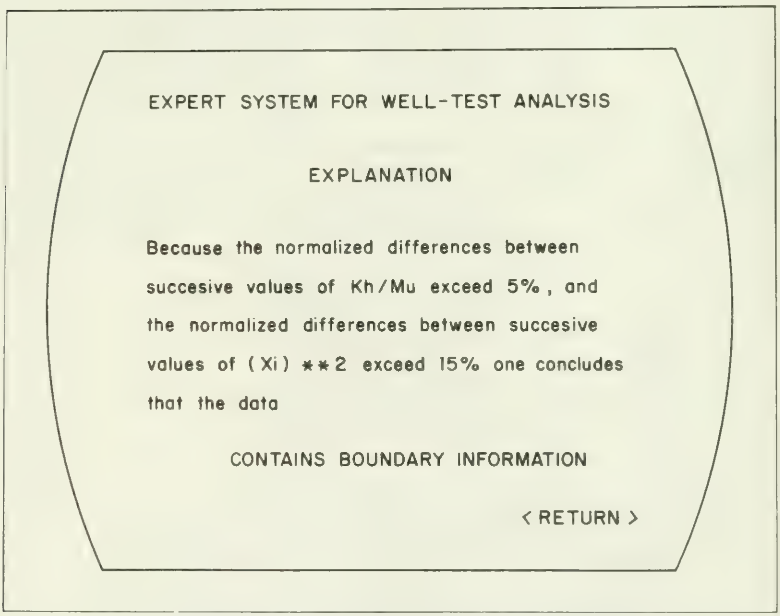


Figure. 5. Example of a screen generated by the display explanations function.

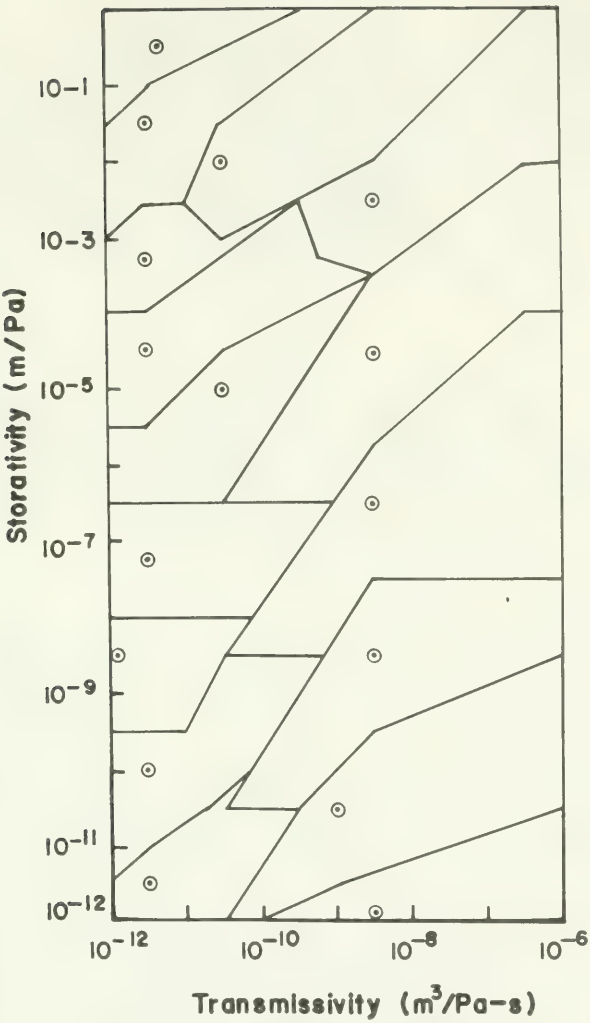


Figure. 6. Trigger points, and the corresponding convergence regions, used by ANAPPRES as initial guesses to start the computational module if the user chooses not to provide initial guesses.

the initial guess provided by the user failed to promote convergence. This map was obtained by the analysis of the solution domain of 230 synthesised well tests.

Inference Engine

This module drives the analysis of the test, on the basis of the options selected and the input data given by the user, the partial results provided by the computational module and the information provided by the knowledge base. The inference engine controls the operation of the computational module, and gets results, such as chi-square and reservoir parameter values, from it. With this information, the inference engine interacts with the knowledge base in order to reach conclusions. Every time the inference engine reaches a conclusion, it commands the user interface to display it in the screen of the VT340 terminal and ask the user if an explanation is requested.

Explanatory Module

It contains preformatted explanations for all the diagnostics and conclusions that ANAPPRES can reach. These explanations are supplemented with information provided by the inference engine each time it reaches a conclusion or diagnostic. If the user chooses to request an explanation, the explanatory module passes the corresponding explanation to the user interface, which displays it through its display explanation function (figure 1).

METHOD OF ANALYSIS

ANAPPRES performs a totally automatic interference tests analysis in a single run. That is, it finds out if there is evidence of a hydrologic boundary in the test data and determines its type, computes storativity and transmissivity values, and distance and angle from an arbitrary origin to the hydrologic boundary.

Figure 7 Illustrates the method of analysis. The analytical process is divided internally into three stages. These are: interference verification, search for hydrologic boundaries and location of hydrologic boundaries. ANAPPRES determines for each observation well the interfering production wells, using superposition. After that, for each observation well ANAPPRES determines if the corresponding data indicate the existence of a boundary, and if there is one, its type (either no-flow or constant-pressure). At this stage, estimates of the storativity and transmissivity associated with the well are obtained; these are taken as final results for the well if no boundary is detected. If a boundary is detected, the values of the transmissivity, storativity and distance of the boundary to the origin are simultaneously determined. This is sequentially done for each and all the observation wells participating in the test.

If two wells detect a boundary, a final analysis simultaneously including both wells is performed to obtain the average values of storativity and transmissivity, the distance of the boundary to the origin and , the angular location of the boundary. However, in

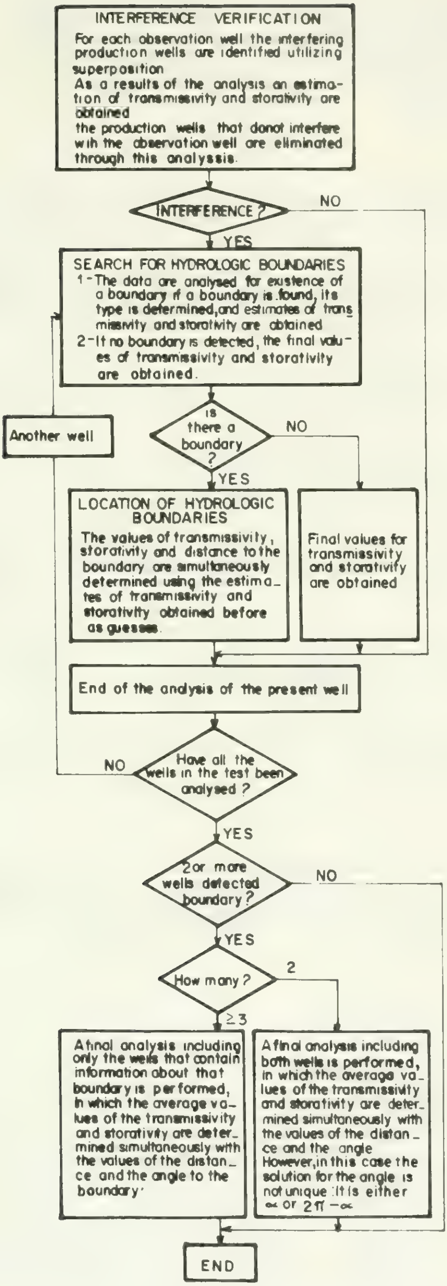


Figure. 7. Method of analysis used by ANAPPRES.

this case θ is not uniquely determined and the true angular location could be $2\pi - \theta$. If more than two wells detect a boundary, a final analysis simultaneously including all these wells is performed to obtain storativity, transmissivity, distance and angle; in this case the angle is uniquely determined.

VALIDATION

At the time of this writing ANAPPRES V2.0 had been validated against a number of published and unpublished problems with known solutions. The former include:

1. A constant-flowrate match of the Theis curve, modeling the simplest interference case in an infinite reservoir [4].

2. A two-well (active and observation), constant-flowrate production interference test in the Raft River, Idaho, geothermal proyect that detected a no-flow boundary [6].

3. A constant-flowrate production interference test between wells 6-2 and 6-1 at the East Mesa geothermal field that detected a constant-pressure boundary [7].

4. A constant-flowrate production interference test between wells 31-1 and 38-30 at the East Mesa geothermal field that detected a no-flow boundary [7].

5. A multiwell, highly-variable flowrate, injection interference test in a shallow groundwater aquifer under consideration for an aquifer thermal energy storage project [4].

6. A multiple-production-well interference test in a geothermal reservoir currently being used for electric power generation [4].

In all cases ANAPPRES obtained the correct diagnoses and quantitative results, whether or not initial guesses were provided by the user. Quantitative results obtained by the expert system agreed with previous results to better than 10 %. These differences are negligible for all practical purposes.

CONCLUSIONS AND FUTURE WORK

We have developed and validated the second version of a computarized expert system capable of analyzing constant and variable flowrate interference tests, in which there are an arbitrary number of production wells and an arbitrary number of observation wells, in liquid-saturated homogeneous reservoirs. The main advantages of this system are that it is user friendly, requires essentially no experience on the part of the analyst, eliminates subjectivity associated with earlier techniques of analysis, can handle complex cases and large data sets, completes the analysis of even the most complex cases (including plotting the results) in one run and is significantly faster than a human expert.

The next version of ANAPPRES, which is already in an advanced stage of development, will include, in addition to the current capabilities, the possibility of analyzing single-well pressure tests.

REFERENCES

1. E. R. Iglesias, V. M. Arellano, M. Castañeda and R. Márquez. "Developments in geothermal energy in México-part eight: geothermal reservoir energy recovery: a 3-D numerical simulation study of the Cerro Prieto geothermal field." Heat Recovery Systems Vol. 6, 1986, pp. 451-463.

2. C. S. Matthews and D. G. Russell. Pressure Buildup and Flow Tests in Wells. Society of Petroleum Engineers Monograph, Vol. 1, 1967.

3. R. C. Earlougher. Advances in Well Test Analysis. Society of Petroleum Engineers Monograph, Vol. 5, 1977.

4. D. G. McEdwards and S. M. Benson. User's manual for ANALYZE, a multiple-well, least squares matching routine for well-test analysis, Report LBL-10907, Lawrence Berkeley Laboratory, 1981.

5. V. M. Arellano. Desarrollo de un Sistema Experto para el Análisis de Pruebas de Presión, M.S. Thesis, Universidad Autónoma del Estado de Morelos, Cuernavaca, Mor., México, 1987, in spanish.

6. T. N. Narasimhan and P. A. Witherspoon. Reservoir Evaluation Tests on RRGE-1 and RRGE-2, Raft River Geothermal Project, Idaho, Report LBL-5958, Lawrence Berkeley Laboratory, 1977.

7. T. N. Narasimhan, D. G. McEdwards and P. A. Witherspoon. Results of Reservoir Evaluation Tests, 1976, East Mesa Geothermal Field, California, Report LBL-5973, Lawrence Berkeley Laboratory, 1977.

Safety Review Advisor

JAMES A. BOSHERS and ISRAEL A. ALGUINDIGUE
University of Tennessee

CATHERINE G. BURNETT
Tennessee Valley Authority

ROBERT E. UHRIG
University of Tennessee

ABSTRACT

The University of Tennessee's Nuclear Engineering department, in cooperation with the Tennessee Valley Authority (TVA), is evaluating the feasibility of utilizing an expert system to aid in 10CFR50.59 evaluations. This paper discusses the history of 10CFR50.59 reviews, and details the development approach used in the construction of a prototype Safety Review Advisor (SRA).

The goals for this expert system prototype are to 1) aid the engineer in the evaluation process by directing his attention to the appropriate critical issues, 2) increase the efficiency, consistency, and thoroughness of the evaluation process, and 3) provide a foundation of appropriate Safety Analysis Report (SAR) references for the reviewer.

---

* Research sponsored by the U.S. Department of Energy

INTRODUCTION AND BACKGROUND

In 1959, the Atomic Energy Commission (AEC), the predecessor to today's Nuclear Regulatory Commission (NRC), issued its first operating license, No. DPR-1, to the Vallecitos Boiling Water Reactor. As issued, the license required that General Electric (GE), the owners and operators, submit for AEC approval, every modification and test or experiment not explicitly approved in the Licensing Documents. These submittals would require AEC approval prior to their implementation. To support the experimental program of the Vallecitos project, GE had to submit several filings a month to the AEC. This arrangement was unacceptable to both the Atomic Energy Commission and General Electric [6].

In 1960, GE asked for, and received a reconsideration of these requirements. GE and the AEC then drafted a new agreement. After formal AEC review, the new agreement was issued as an amendment to DPR-1 on a memorandum and order dated 2 November, 1960. The amendment clearly stated that GE had complete freedom to make changes within the parameters of the technical specifications, provided that no unresolved safety question was involved.

Recognizing the widespread applicability of this approach to regulating changes to licensed facilities, the AEC issued proposed rule 10CFR50.59. The purpose of this new rule was to define the extent to which the licensee could make changes, and perform tests or experiments that were not specifically allowed for in the operating license. Four months after it was proposed, Title 10 of the Code of Federal Regulations part 50 section 59 (10CFR50.59) became effective on August 9, 1962.

Today, all licensed nuclear facilities are subject to 10CFR50.59. This regulation is valuable both to the licensees and to the NRC. For the owners/operators, it allows the freedom to operate and control their facility. The NRC finds the regulation valuable because it maintains the original licensing basis of the facility. Nevertheless, the implementation of this regulation has caused a great deal of confusion, both with the licensees and the NRC [6]. The difficulty comes in the interpretation of the document and the implementation of its requirements.

Specifically, 10CFR50.59 permits the licensee to make changes to the facility or procedures, and to conduct tests or experiments without prior NRC approval provided that the change, test, or experiment meets certain criteria. These criteria are:

1) The proposed activity must not involve a change in the technical specifications and,

2) The proposed activity must not involve an unreviewed safety question.

The first of these two criteria is rather straight forward and redundant since NRC approval is required for all technical specification changes. The difficulty comes in the interpretation of the second. The definition of an "unreviewed safety question" provided by the NRC in 10CFR50.59 is stated as follows:

10CFR50.59 (2)

A proposed change, test, or experiment shall be deemed to involve an unreviewed safety question if;

i) the probability of occurrence or the consequences of an accident or malfunction of equipment important to safety previously evaluated in the Safety Analysis Report may be increased or,

ii) a possibility for an accident or malfunction of a different type than any evaluated previously in the safety analysis report may be created or,

iii) the margin of safety as defined in the basis for any technical specification is reduced [2].

An engineer attempting to evaluate a proposed modification, test or experiment must first determine several other issues. Questions like

1) What equipment is "important to safety"?

2) What is the Safety Analysis Report?

3) What is meant by "evaluated previously in the safety analysis report"?

4) What is a "margin of safety as defined in the basis for any technical specification"?

must be answered before the engineer may proceed [6]. The answers to these questions may differ depending on the facility and utility to which they are addressed. For example, the SAR is a broad term that can refer to an entire group of documents that were submitted to the NRC for approval. It usually includes the Final Safety Analysis Report (FSAR), the technical specifications, and the basis for the technical specifications, but it can also include design bases and design criteria documents. The SAR also includes all commitments documented in Safety Evaluation Reports (SER). The SER is a document written by the NRC to support the issuance of the operating license and is based on information provided in the SAR. SERs are also written on subsequent submittals that may revise the operating license.

A successful application of the 10CFR50.59 criteria to a proposed change, test or experiment requires that the individual performing the review be knowledgeable of the licensing documents associated with the particular facility, as well as the facility and especially all its safety related systems. This requires an amount of expertise that usually precludes any one individual from being knowledgeable in all areas. Thus in most cases, the engineer performing the review must consult with several other engineers to properly evaluate the proposed activity.

If a 10CFR50.59 evaluation is done improperly, the results can be very serious. Aside from fines that could be imposed, there is a potential for the creation of a real threat to the safety and health of the public that could go undetected.

PROJECT APPROACH

The nature of safety reviews requires that individuals performing these evaluations be knowledgeable in many areas of engineering. Evaluators must be familiar not only with licensing documents, but also with the facility, and specifically with safety related systems. Clearly, the amount of information to be processed is considerable. Hence, an expert system can significantly assist in the preparation of these reviews.

Expert Systems

An expert system is an application program that attempts to mimic human judgment by applying substantial knowledge of specific areas of expertise to solve finite, well-defined problems. By capturing in computer code the expertise of highly qualified individuals, problems which reside in the same domain can be solved by repetitively applying the same knowledge.

An expert system typically consists of two components, an inference engine and a knowledge base. The inference engine gathers information, conducts searches, and draws inferences based on the strategy programmed into it. Once conclusions have been secured, recommendations are presented along with explanations on the bases for the conclusions.

The knowledge base of an expert system contains the expertise - collected from experts, books and publications - used in providing advice under a variety of conditions. This expertise describes a methodology for solving a problem as a human expert would solve it. The knowledge is encoded using rules, frames, or other techniques for knowledge representation and is manipulated by the inference engine to provide advice and recommendations.

There are some benefits to be obtained from the use of expert systems. Expert systems make it possible to deliver expertise to all locations at all times. It is also apparent that aside from providing advice, expert systems become repositories for undocumented knowledge which could otherwise be lost through retirement.

Another obvious advantage is that expert systems do not become fatigued. Expert systems can provide objective, consistent expert advice and rapid access to databases and other vital information.

Expert Systems and Conventional Computer Programs. The basic difference between expert systems and conventional computer programs is that expert systems manipulate knowledge while conventional programs manipulate data [5]. In a

conventional computer language, instructions to be executed are presented sequentially and are highly interconnected. There is an algorithm to be followed, and execution of the program implies a logic flow from one instruction to the next as presented in the code sequence. The expert system style of programming has fundamentally changed the way we give instructions to the computer and how the machine executes those instructions. Instructions are logically connected - not sequentially - and as long as there is a logical link between the input and the conclusions, the inference engine will eventually arrive at a result.

The separation of knowledge and inference techniques in an expert system simplifies greatly the task of updating knowledge bases. Since knowledge is structured independently, it remains distinct and legible and may be deleted, changed or included in a system without extensive logic redesign. In conventional computer programs, in contrast, the knowledge is interwoven with the program logic and structure, and changes are bound to disturb the behavior of the program.

An expert system contains a degree of self-awareness or self-knowledge that allows it to reason about its own operations. This self-knowledge gives an expert system the ability to provide explanations on its decisions and to generate status determination information.

Expert systems also have the ability to manipulate uncertain, or fuzzy data. When the data in the knowledge base is specific and precise, expert systems give results that are unambiguous. However, when information is not precise, incomplete, missing or conflicting, expert systems can still reach a rational conclusion or solution through the use of confidence factors. Under these conditions, an expert system will give the "most probable" solution or the "best" solution, but not necessarily the correct solution [3]. Experts are sometimes forced to make subjective evaluations. Such subjectivity may be easily incorporated into an expert system using confidence factors.

Applications of Expert Systems. The impact of the technology of expert systems has been felt in many areas of science, education, and industry. In the last ten

years, development efforts have resulted in the implementation of a great number of applications now operational, or in the prototype stage. In the nuclear industry, there are many areas in which expert systems could make significant contributions. Expert systems are foreseen as providing promising solutions for problems in personnel training, plant management and safety evaluation.

The ability of expert systems to generate status determination information and to provide explanations on the bases of their conclusions, can be used in the training of personnel [4]. Additional benefits are to be gained from the design and implementation of the system itself, which will force the development and documentation of decision making policies.

In the management and operation of nuclear power plants, expert systems can contribute as expert assistants to the operators, as monitoring and validating systems for sensor data, and as on-line access system for performance and safety data. Obviously, the robustness and completeness of the systems to be developed is of critical concern.

The Safety Review Advisor

The Safety Review Advisor is an expert system to aid in 10CFR50.59 evaluations. In building the SRA, we attempted to emulate the thought process of a reviewer. To accomplish this, the expert system must ask some questions that the engineer would address automatically. For example, the engineer evaluating the proposed activity must first determine whether the activity is a change, or a test or experiment and then apply the appropriate criteria.

Once this distinction has been made, the engineer begins to evaluate the proposed activity in greater detail. If the proposed activity is a change, the engineer must determine whether the change will affect only the facility or will also require a change to a procedure. However, both possibilities must be evaluated since either could require NRC approval prior to implementation.

Figure 1 shows the block diagram of the logic used in our approach to solving the problem. Since our efforts are presently directed towards the construction of a proof of principle prototype, the SRA deals exclusively with changes to the facility. Specifically, the prototype addresses changes directly affecting the Standby Power system of the Sequoyah Nuclear facility. Figure 2 shows the block diagram of the prototype under construction.

A full scale system, as depicted in figure 1, would address changes to both the facility and procedures, as well as the possible effects of proposed tests or experiments. Our choice of which branch to model was based on conversations with TVA engineering personnel. They indicated that the majority of their evaluations, and the ones with the most potential impact on safety, were performed on proposed changes to the facility. The Standby Power system was chosen because it has very few systems which support it. The few number of support systems allowed for a more rapid development of the prototype.

To evaluate a proposed change to a facility, an engineer must determine which of the safety related system or systems would be directly affected by the change. The engineer must also determine the possible effects the change would have on other systems that interface with it.

Our prototype takes a similar approach. The SRA first determines the system on which the proposed change will be performed. The evaluator selects the appropriate system from a list of plant systems. A full scale version of this system would allow the engineer to choose more than one of the systems listed. Proposed modifications may directly affect several systems. The prototype on the other hand, only allows the selection of the Standby Power system.

The SRA then attempts to narrow the scope of attention by inquiring on the subsystem that the modification will affect. In the Standby Power system there are two major subsystems. These are the Emergency Standby AC Power Supply and the Vital DC Power Supply. The reviewer must then select one of the two. Once the appropriate subsystem is chosen, the SRA must determine which components or sets of components will be affected by the modification. Each inquiry presents a different list of choices based on the reviewer's input.

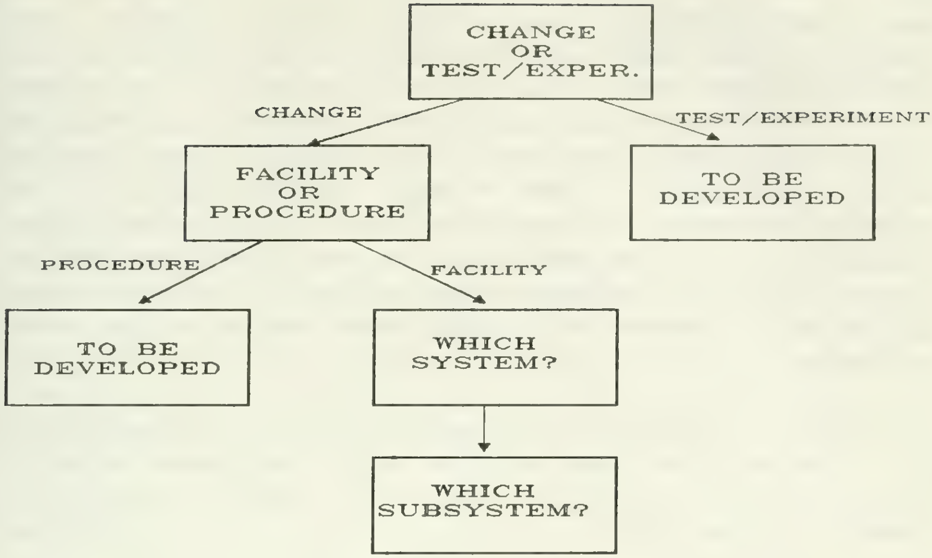


Figure 1. Full-Scale System Block Diagram.

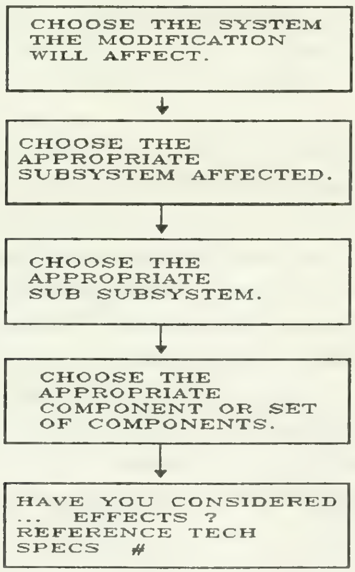


Figure 2. The SRA Block Diagram.

After the prototype sufficiently narrows the focus to a particular component or group of components, the system responds with the appropriate information. This information includes design criteria references, SAR references, and other critical information ("rules of thumb") relevant to the component. For instance, if the reviewer were evaluating the possible modification of the diesel generator's heat exchanger, the prototype would list all the appropriate SAR references that address the heat exchanger. The SRA would provide the appropriate requirements set forth in the design criteria and design basis documents, and it would also provide rules of thumb used by the diesel generator experts in evaluating the performance of the component (such as "only 15% of the tubes in the heat exchanger may be plugged without affecting its heat removal requirements").

The collection of information presented to the evaluator is the biggest benefit of using the SRA. This information contains not only the expertise of several experts, but also references which could be very helpful in ensuring and documenting a complete review.

TVA, like many other utilities, has designated engineers to perform Safety Evaluations. Easy access to information about each system and their major components, would significantly reduce the research time required to perform the evaluations. It would also tend to make safety evaluations more consistent.

PROJECT STATUS

Development of the SRA began in January 1989 and it is approximately one third complete. The current version of the system was coded in Texas Instruments' PC Plus following a modular design plan. Eventually each module of the SRA will address a different plant system. Interconnections among systems are also included in the design. Once completed, the SRA will be capable of addressing all modifications to Sequoyah's Standby Power system.

The feasibility of incorporating simplified plant system schematics into the prototype is being explored. Development to date has indicated that this capability will be a necessity for implementation of a large scale system.

RECOMMENDATIONS

If the system is to be expanded, it would be advisable to transport the system to a more powerful expert system environment or to code the SRA directly in a programming language. The size of a full-scale system is assured to cause a significant increase in the time required to run a consultation under PC Plus.

One of the major issues of a full sized expert system would be its ability to incorporate drawings of plant systems and schematics. Inclusion of plant drawings could aid the transfer of information from the SRA to the reviewer. However, including this feature would make development of the SRA more complex. Another issue for a full size SRA is the amount of effort that it would be necessary to maintain it up to date. Each change to the facility would have to be reviewed for possible impact on the SRA.

CONCLUSIONS

The prototype Safety Review Advisor under development will address proposed modifications to the Standby Power system. The prototype will not provide a perfect evaluation of a proposed modification. It cannot address all possible changes, tests or experiments. The best that can be hoped for with the technology at hand is to develop an aid for the engineer to help him perform the evaluations more thoroughly, consistently and efficiently.

If the SRA were to be developed on a full scale, the issue of the engineer's dependency on the system would have to be addressed. The prototype is designed strictly as a tool to assist engineers in performing the evaluations. As with any other tool, the old adage " you need to be smarter than the machine you are trying to operate " still applies. There is concern that such a system would create complacency in the engineer. An engineer using an expert system must always be aware of the boundaries and limits of such a system. The goal of the system is to direct the reviewer's attention to potential areas of concern, not to perform the evaluation.

REFERENCES

[1] Burnett, Catherine G., "Functional Specifications Safety Review Advisor (SRA)," Tennessee Valley Authority, Division of Nuclear Engineering, Engineering and Computer Methods Branch, Internal Memorandum.

[2] Title 10, U. S. Government, Code of Federal Regulations, Part 50, Section 59.

[3] Uhrig, Robert E., "Toward the Next Generation of Nuclear Power Plants," Forum for Applied Research and Public Policy, Fall 1986.

[4] Uhrig, Robert E., "Artificial Intelligence and Training of Nuclear Reactor Personnel," Proceedings of the CSNI Specialist Meeting on Training of Nuclear Reactor Personnel, Orlando, Florida, April 21-24, 1987.

[5] Waterman, Donald A., "How do Expert Systems Differ from Conventional Programs," Expert Systems, Vol. 3, No. 1, January 1986.

[6] Williams, Daniel H., "10CFR50.59: Why We Have it - What it Says," Personal Communication.

CEXS: An Expert System for Corrosion Monitoring in Nuclear Power Plant Service Water Systems

LYDIA B. BROWN
Odetics, Inc.
1515 South Manchester Avenue
Anaheim, California 92802, USA

ABSTRACT

A major concern in the nuclear industry today is the corrosion of plant piping and components in the Service Water System. Untimely failures cost plants millions of dollars in down time and possible safety violations. The corrosion problem differs from reactor to reactor, depending on source of cooling water, EPA requirements for dispatch, system design and construction materials, and treatment methods. Still, a method is needed for predicting possible component failures due to corrosion so they can be replaced during a scheduled outage. The CEXS expert system advises the plant technical staff of predicted component degradation and possible failures. Direct monitoring of components occurs continuously and the expert system advises the user when maintenance or replacement should be scheduled. CEXS is implemented on a PC which is powerful, portable, and contains a graphic interface that is easy to use.

INTRODUCTION

Problem Definition

A growing concern among nuclear power plants is the rapid corrosion of piping systems and components, especially those which are subjected to untreated, stagnant, or organism-infested water. Of special concern are Service Water Systems, which are more likely to experience these conditions than other piping systems. Components designed to last the life of the reactor are failing long before their forty year life expectancy, or degrading so quickly that failures are expected before the plant has reached an age of twenty. Because the source

of service water differs from utility to utility, each plant will have different kinds of problems depending on the corrosive elements, microbes, suspended silt particles, animal life, etc., in the water. Restrictions on chemical additives used to reduce corrosion, the materials used in the piping and components, and the flow path also affect the type and severity of the corrosion problem. Often the problems are compounded by the interactions of corrosion, erosion, and blockage.

Increasing the urgency of finding a solution is the fact that many of the components affected by untreated water are safety-related and would require the eventual shutdown of the plant if they were to fail.

It is important to be able to predict these failures before they occur, since it is impossible in most cases to prevent them, due to ecological limitations on chemical treatment of water released to the environment (open loop systems), uncertainty as to the cause(s), and design limitations. An expert system which monitors data from the system components and piping in order to make these predictions and advise the chemistry staff, would reduce unscheduled down time by allowing plant technicians to maintain or replace components at scheduled outages before the components are expected to fail. The expert system would not only provide the user with knowledge and reasoning ability based on human experts, but ensure that this knowledge and reasoning capability would be available even when the human experts are inaccessible.

The Corrosion Expert System (CEXS) is a prototype that demonstrates the feasibility of the development of a full-scale expert system. It contains a user-friendly graphics interface which allows quick access to, and efficient use of, the information available through CEXS. The system is easy to use and requires very little time for familiarization.

Economic Need

Service Water Systems have traditionally attracted less attention than other nuclear plant systems until recently, when it began to be noticed that corrosion effects in Service Water Systems caused much of the unscheduled down time. Untreated water in many systems accelerates corrosion, so that many nuclear power plants which are approaching middle age are experiencing an increasing number of

component failures. Newer plants are predicting that some of their components will not even last to middle age. North Anna, located in Virginia, had such severe corrosion that it was estimated (in 1981) that in three to six years, the entire Service Water System would need to be replaced. North Anna initiated an extensive cleaning, treatment, and monitoring program designed to reduce the corrosion to a level that would ensure system operability for the 30 remaining years of plant life. At this time, North Anna is achieving acceptable corrosion rates, but continued long term monitoring is necessary to make sure corrosion rates remain in an acceptable range.

Safety Need

The U.S. NRC Office for Analysis and Evaluation of Operational Data has issued a report entitled "Service Water System Failures and Degradations in Light Water Reactors." The report indicates that Service Water System failures and degradation have significant safety implications. Of the 276 off-normal events examined, a major cause of degradation was corrosion and erosion. Corrosion and erosion accounted for 77 events, or 28% of the failures and degradations examined.

Corrosion Monitoring of Service Water System Components

Corrosion is monitored in two ways: continuously, by corrosion monitors (devices that measure corrosion electrolytically); and periodically, by corrosion coupons (samples of the alloys used in the components and piping) which are inserted directly into the Service Water. The corrosion coupons are time-consuming to use and provide data only every 30-90 days when measurements are conducted. They are an accurate measurement of the corrosion since they are constructed of the same materials as the components and piping (mild steel and a copper/nickel alloy for the Zion Service Water System), and are strategically placed near the actual components. Data from the coupons offer a correction to the instantaneous corrosion rate which is always available, although may not be as accurate. Since the coupon corrosion rates are measured by hand, they must also be entered into the expert system manually. CEXS provides the user with a pull-down menu and edit box to facilitate the data entry.

SOLUTION APPROACH

Scope of Prototype

The prototype CEXS was built using Zion Station's Service Water System as a model. Commonwealth Edison's Zion Station, located in Illinois, uses untreated Service Water from Lake Michigan in a once-through loop, and therefore, has more immediate corrosion problems than a system that uses purified, treated water or a closed loop system.

Because Service Water is discharged to Lake Michigan, Zion must follow strong environmental guidelines restricting chemical treatment of the water. Also, if the Service Water flow rate is low, suspended silt particles will settle out. These depositions of silt make ideal protected crevices for corrosive agents. When the silt was cleaned from one Zion Station heat exchanger, it developed many leaks; the silt had been plugging the holes.

These observations show Zion's immediate need for an expert system that monitors corrosion and makes it an ideal choice for use in a prototype system.

Research was confined to the Zion Component Cooling Water Heat Exchangers for several reasons: they have the largest heat removal load of any heat exchanger supplied by Service Water, and they are safety-related components. In addition, if a Component Cooling Water Heat Exchanger were to fail, water could leak from the Service Water side to the Component Cooling Water side, contaminating a relatively "clean" system with corrosive agents. This would increase the corrosion rate in the Component Cooling Water System where significant amounts of corrosion are unacceptable. If water were to leak in the other direction (i.e., from the Component Cooling Water System to the Service Water System), radioactive materials and the chemicals used to treat the Component Cooling Water would be released into Lake Michigan at high economic cost for environmental clean-up and possible fines.

System Design

Hardware. A Portable Compaq 386/20 with 6 Mbytes of expanded memory was selected as the platform for CEXS because of its portability, small space requirements,

and low cost. The supporting hardware consists of a NEC Color Monitor, used to display the graphics, and a Microsoft Mouse, which is the primary means of communication between the user and CEXS. The Compaq 386 is fully IBM-compatible and runs MS-DOS.

<u>Graphic Interface</u>. One of the goals in designing this expert system was to ensure that the end user would not have to be a programmer in order to use the system effectively. For this reason, graphics were used to design a man-machine interface that could effectively be used by plant technical personnel and the chemistry staff in a real world environment. The user interface to this Corrosion Expert System has three main characteristics:

- It provides a friendly graphics system interface with a familiar screen layout, excellent display techniques and easy user interaction capability.

- It alerts the user to important messages through the use of audio beeps as well as screen text.

- It utilizes color and takes into account human factors so that the plant technical personnel and the chemistry staff can readily understand the system operation.

The user interface is the only means of interacting with the expert system and the CEXS operation procedure; therefore, we have designed a system emphasizing a graphics interface. "Mouse sensitive" objects can provide additional information about a specific component when the mouse button is activated.

<u>Use of Software Packages</u>. Nexpert Object from Neuron Data, Inc., was chosen as the expert system shell to develop CEXS for the following reasons:

- Nexpert is portable to many other hardware platforms, including the MacIntosh, SUN and VAX workstations.

- Nexpert handles both object and rule representations.

- Nexpert reasons both forward and backward and provides the inferencing mechanism needed in an expert system.

Microsoft Windows was also chosen for its ease of operability by the user, and because it had the ability to meet the criteria we had established for a graphics interface. Windows is an operating environment which sits on top of the operating system, MS-DOS, enhancing but not eliminating it. It provides extensive graphic capability, multitasking, and mouse input as well as keyboard input. Microsoft Windows offers the user limitless flexibility in graphics design within the framework of a standardized window display, menu, and dialogue box system. Thus, the user's task in becoming familiar with the system is simplified: the computer graphics representation of the components is easy to understand, and the control menus are like those of other standard windowing systems. Because it is a multitasking environment, Windows is able to run several applications at one time, and with less RAM memory than would normally be required. It automatically loads and unloads segments of code that it needs or doesn't need at any particular time. Windows graphics are also device-independent, meaning that the same subroutine calls that are used to draw graphics on the monitor screen can be used to print out on paper.

<u>Program Control Flow</u>. Figure 1 shows the modules in CEXS and their interrelationships to each other and to the Windows and Nexpert software packages. Each module performs only one function; in this way CEXS maintains program modularity, making it easy to modify any part of the expert system as the needs of the user expand and change.

The primary method of communication between Windows, CEXS and Nexpert is the message. Messages, rather than procedures, control the flow of the program and allow true user interaction. For instance, a mouse-click by the user creates a message in Windows, which sends the message to the proper CEXS routine to be processed. If CEXS has no special action it wants to take, CEXS sends the message back to Windows to be processed by Windows in the default manner for that message. Otherwise, CEXS processes the message itself. Another example of program control by messages is CEXS's interaction with the reasoning power of Nexpert. When CEXS calculates current information from the data, it sends a message to Nexpert, which reasons on the new information. Nexpert, in turn, calls routines in CEXS to act on its reasoning. These examples are generalized in the program control flow by messages. (See Figure 2).

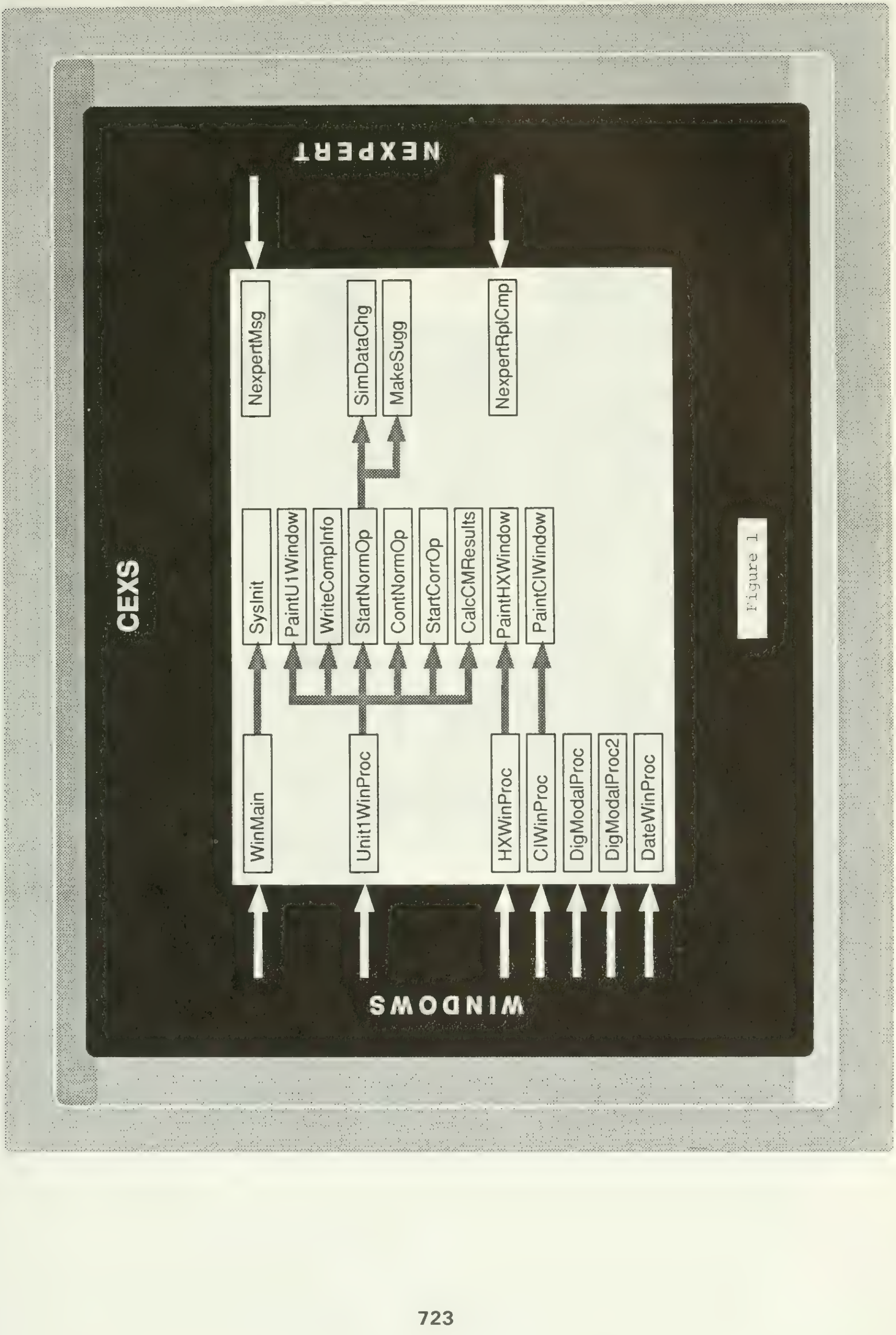

CEXS
NEXPERT
WINDOWS
NexpertMsg
SimDataChg
MakeSugg
NexpertRplCmp
SysInit
PaintU1Window
WriteCompInfo
StartNormOp
ContNormOp
StartCorrOp
CalcCMResults
PaintHXWindow
PaintClWindow
WinMain
Unit1WinProc
HXWinProc
ClWinProc
DigModalProc
DigModalProc2
DateWinProc

Figure 1

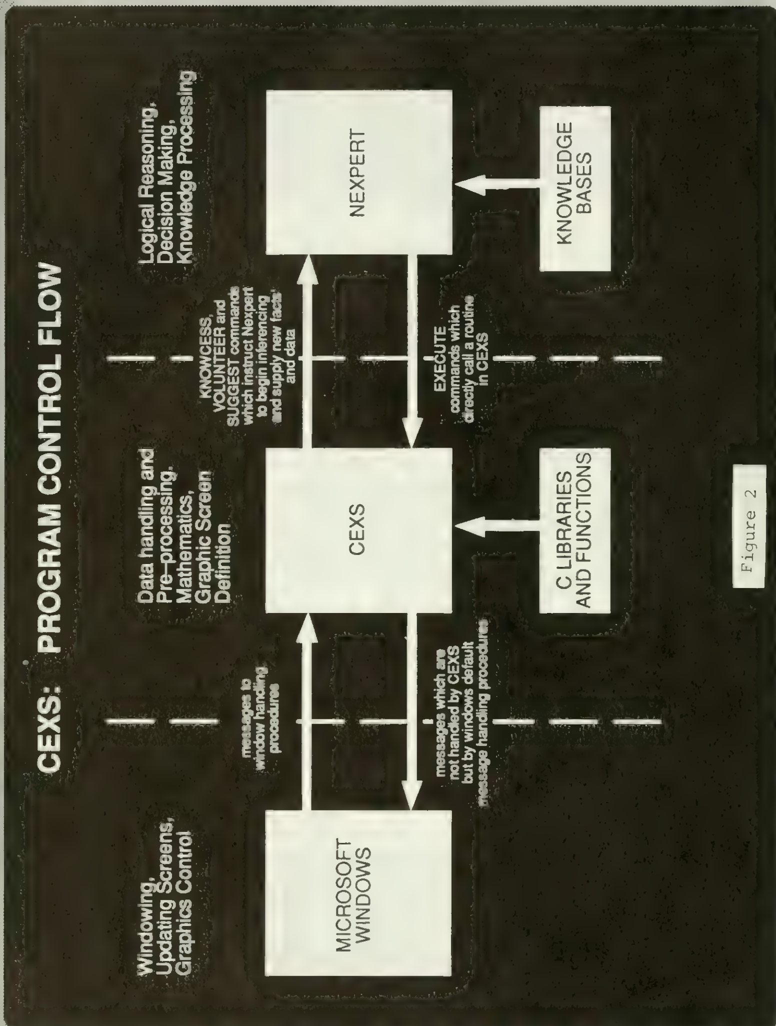

CEXS: PROGRAM CONTROL FLOW
Windowing, Updating Screens, Graphics Control
Data handling and Pre-processing, Mathematics, Graphic Screen Definition
Logical Reasoning, Decision Making, Knowledge Processing
MICROSOFT WINDOWS
CEXS
NEXPERT
messages to window handling procedures
messages which are not handled by CEXS but by windows default message handling procedures
KNOWCESS, VOLUNTEER and SUGGEST commands which instruct Nexpert to begin inferencing and supply new facts and data
EXECUTE commands which directly call a routine in CEXS
C LIBRARIES AND FUNCTIONS
KNOWLEDGE BASES

Figure 2

Estimate of the Failure Date for Components. The estimated failure date for major components is predicted by CEXS using the current corrosion rate and the rate of change of the corrosion rate. Thus, if the corrosion rate is increasing rapidly, CEXS predicts that the corrosion rate will continue to rise and the end result will be a shorter component life. If, instead, the corrosion rate is relatively constant, CEXS predicts that the corrosion rate will be stable. These are crude assumptions and will be revised in time as more is understood about how corrosion rates vary with time.

Dual-Mode Operation. The first mode operates without corrosion monitoring to show the current situation at Zion Station. The second mode demonstrates the same scenario with the proposed corrosion monitoring approach to assess the benefit of corrosion monitoring using CEXS. The first mode, Normal Operations, which simulates Zion Station as it currently exists, does not provide the user any corrosion data on any of the components. The simulation runs smoothly until a Component Cooling Water Heat Exchanger fails suddenly and without warning, resulting in unscheduled down time at a potential cost of millions of dollars in lost power. The second mode, Corrosion Monitoring, is really a demonstration of CEXS itself. (The production version of CEXS would not include a Normal Operations mode; it was meant only for demonstration purposes to show the effectiveness of corrosion monitoring devices.) Corrosion Monitoring tracks all the component data, calculates an estimated failure date due to corrosion and decides at which outage the component should be serviced. A message appears on the screen to announce the occurrence of a scheduled outage. For demonstration purposes, the component is assumed to be maintenanced during the outage CEXS advises. The new estimated failure date is much farther in the future. We have also included thermal performance monitoring for the Component Cooling Water (Service Water System) Heat Exchangers in order to allow the user to examine corrosion effects based on thermal performance.

PROJECTED DEVELOPMENT

Future development of the Corrosion Expert System will include the estimation of the erosion effects on major components in the Service Water System (SWS), and the expansion of corrosion monitoring to include all major components of the SWS at Zion Station. Erosion estimation will be done using a flow model of Zion's SWS.

The user will be notified of excessive corrosion, erosion, and total component degradation rates, and when thermal performance is approaching design specifications.

Graphs of the following will be plotted for each major component:

- Corrosion rate v. time

- Erosion rate v. time

- Total component degradation rate v. time

- Thermal performance effectiveness v. time (heat exchangers only)

- Heat transfer coefficient v. time (heat exchangers only)

In order to make the man/machine interface more natural for the user, CEXS will include the capability of editing and changing manually-entered data.

The technology developed for this project also has applications to other piping systems and components that are affected by corrosion, erosion, and degraded thermal performance.

SUMMARY

A functional, demonstrable prototype system for monitoring corrosion was developed which:

- monitors data from simulated corrosion monitor devices and coupons strategically placed throughout the Service Water System.

- analyzes the data and information based on the expert knowledge provided.

- provides guidelines for key component repair or replacement. Predictions of when these components will fail are based on the simulated data and the knowledge acquired from experts.

- decreases unscheduled down time due to component failure because of corrosion effects in the Service Water System.
- uses graphics, color and a mouse to provide a man/machine interface that is simple to use.
- contains an architecture that can be used as a basic structure for the future product development.

A Decision Support System Based on Hybrid Knowledge Approach for Nuclear Power Plant Operation

JOON ON YANG and SOON HEUNG CHANG
Department of Nuclear Engineering
Korea Advanced Institute of Science and Technology
P.O. Box 150
Cheongryang, Seoul, Korea

ABSTRACT

This paper describes a diagnostic expert system, HYPOSS (Hybrid Knowledge Based Plant Operation Supporting System), which has been developed to support operators' decision making during the transients of nuclear power plant. HYPOSS adopts the hybrid knowledge approach which combines shallow and deep knowledge to couple the merits of both approaches. In HYPOSS, four types of knowledge are used according to the steps of diagnosis procedure : structural, functional, behavioral and heuristic knowledge. Frames and rules are adopted to represent the various knowledge types. Rule-based deduction and abduction are used for shallow and deep knowledge based reasoning respectively. The event-based operational guidelines are provided to the operator according to the diagnosed results. If the exact anomalies can not be identified while some of the critical safety functions are challenged, the function-based operational guidelines are provided to the operator. For the validation of HYPOSS, several tests have been performed based on the data produced by a plant simulator. The results of validation studies showed a good applicability of HYPOSS to the anomaly diagnosis of nuclear power plant.

1
INTRODUCTION

The need to use computers in Nuclear Power Plants (NPP) by the operating crew as an aid in making decisions has been widely endorsed in the numerous investigations following the Three Mile Island Nuclear Station Unit 2 (TMI-2) accident in March 1979. The use of computer is expected to provide information analysis and intergration of functions not achievable with the conventional control room instrument.

A post TMI-study concluded that [1]:

1) Time stress due to information overload and decision uncertainty increase the risk of error.
2) Even with better training and improved procedures, human error can never be fully eliminated.
3) Computer-assisted support systems are a promising technology innovation to aid knowledge-based decision making by providing processed and derived information formed to assist the cognitive process, reducing information overload and thereby stress, reinforcing decision skills and thereby increasing confidence, and providing an overview of plant status to facilitate the prompt detection of inevitable errors.

A number of systems have been developed as computerized decision aids: alarm analysis systems, disturbance analysis systems (DAS), safety parameter display systems (SPDS), disturbance analysis and surveillance systems (DASS), alarm handling systems and expert systems. The detailed contents of them are presented in many works [1-3].

Among them, artificial intelligence (AI) in the form of expert systems is being considered for a variety of operator support functions. Since REACTOR was proposed by Nelson in 1982 [4], a number of expert systems have been developed for operational assistance [5-14].

In general, the knowledge used in expert systems can be divided into two types : shallow and deep knowledge. Shallow knowledge is the knowledge with no explicit

representation of the underlying principles, e.g. cause-consequence tree, statistical result and heuristics. Deep knowledge explicitly represents the underlying physical principles [10].

Most current applications of diagnostic expert systems in electrical and medical domain are built based on shallow knowledge, e.g. rule-based approach. They utilize simple production rules to provide a mapping between possible faults and signals from system. The result is an expert system with impressive capabilities within the area of expertise for which it has been prepared [15]. Some diagnostic expert systems for NPP used this approach. RSAS developed by Sebo used production rule-based approach for use at NRC Operation Center [9]. COPILOT developed by Kaplan used Bayes' theorem to identify the cause of reactor trip [13] and Erdmann used fault tree to build an expert system for safety diagnosis [14].

However, most diagnostic expert systems for NPP utilize deep knowledge approach [6]. This is due to the fact that the diagnosis of anomalies of NPP is different from that of the electrical or medical domain. The differences are [5]:

1) The system is very large and complex, i.e. the system consists of a large number of components, lines, valves and etc.
2) The system is dynamic, i.e. the observable signals are time dependent.
3) Many of important signals are observable.

For instance, Yamada and Kiguchi attempted to use structural and causal knowledge for the accident diagnosis of NPP respectively [5,7]. Nelson developed the response tree method used in REACTOR [4,6]. Washio used semantic net to represent the structure of NPP [8]. Herbert applied a method of qualitative physics called IQA to the failure diagnosis of pressurizer [10]. Guarro implemented the logic flowgraph which is similar to semantic net [12]. Also, some systems used transient analysis code to simulate the consequences of generated hypotheses [11].

However, when one tries to build a practical diagnosis system, several limitations of deep knowledge based reasoning emerge. One is the fact that not all tasks needed for a practical system are easily accomplished with today's understanding of deep knowledge approach. Hence, a body of work is emerging on the appropriate coupling of deep knowledge of the system with shallow knowledge [15-16].

In this consideration, we have developed an expert system, HYPOSS, to diagnose the anqmalies of NPP and to offer correct operational response guidelines using hybrid knowledge approach which combines shallow and deep knowledge to couple the merits of both approaches. In HYPOSS, at the initial stage of transients, shallow

knowledge based reasoning is applied. When the anomalies can not be diagnosed by the shallow knowledge based reasoning, deep knowledge based reasoning is applied. Heuristic rules are used for shallow knowledge based reasoning while structural, functional and behavioral knowledge is used for deep knowledge based reasoning. The rule-based deduction and abduction are used for shallow and deep knowledge based reasoning respectively as inference strategies.

2
ORGANIZATION OF HYPOSS

The overall structure of HYPOSS is shown in Fig. 1. The system consists of one input processor, six knowledge bases and three data bases. The input processor transforms the numerical data from NPP into symbolic form as used in HYPOSS. This part, also, calculates the quantities such as enthalpy and the change of inventory,etc., which are not obtainable directly from NPP, using the signals and design data of NPP. The knowledge base is divided into three parts : the first one is for deep knowledge, the second is for shallow knowledge and the last is for emergency operation guidelines. The name of each knowledge and data base implies the content of it. HYPOSS is built on IBM-PC AT Compatible using SD-Prolog [17].

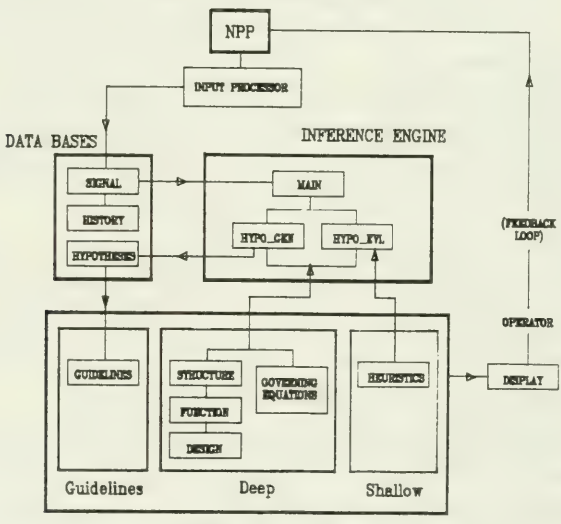


Fig.1 Overall Structure of HYPOSS

3
KNOWLEDGE BASE OF HYPOSS

3.1 KNOWLEDGE BASE FOR DEEP KNOWLEDGE

There are three general types that deep knowledge may take - structural, functional and behavioral. Structural knowledge consists of the physical relationships among the parts of a system, commonly called connectivity, and the manner in which the individual parts of a system are constructed. Functional knowledge is related to the idea of intended structure. And behavioral or causal knowledge represents the knowledge of how the parts of a system behave.

The separation and combinative use of these knowledge are major characteristics of HYPOSS. So, the knowledge base for deep knowledge is divided into three parts : for structural, functional and behavioral knowledge. In this section, the principles to construct deep knowledge base will be presented.

3.1.1 Structure and Function Representation

A particular system works on certain underlying principles. Due to the components that make it up and how these components are connected, the system manifests a certain behavior. The connectivity and functionality of the system constitute some of the fundamental knowledge that a human acquires all other knowledge about diagnosing the system. Regardless of the ultimate functionality of a system, most are built based on a finite set of fundamental building blocks. These building blocks represent some general functionality that can be found within many different systems. Based on this notion of basic building blocks, a set of "fundamental primitives" was developed to describe to the computer how a specific system functions [16].

The set of fundamental primitives for structure description that we have implemented in our work is 1) "components", 2) "lines" and 3) "equipments".

1) Components are used to describe the passive objects which consist of vessel such as reactor, pressurizer and steam generator, etc.

2) Lines describe the connections between components.

3) Equipments represent the active objects such as various pumps,valves and heaters, etc.

The typical forms of component and line representations are shown in Fig. 2. The equipment representations are included in component or line representation.

```
component(level(2),class(1),name(prz),
          mass_in([prz_surge,prz_spray]),
          mass_out([prz_out]),
          energy_in([prz_heater]),
          energy_out([])).
line(level(2),class(1),name(charging),
    from(rwst),
    to(prz),
    through(pump([charging_pump]),
          valve([charging_valve]),
          heater([]))).
```

Fig.2 Typical Form of Structure Representation

For function description, the functions of NPP are classified into the five types as given below.

1) power control function
2) pressure control function
3) inventory control function
4) flow rate control function
5) temperature control function

The other functions such as chemical control are not considered in the present stage. The five types of functions considered in HYPOSS imply that the transients of NPP are, also, standardized as five types related to each function. The typical forms of function representation are shown in Fig. 3.

The structure and function are represented using frames. The level and class in Figs. 2 and 3 represent the hierarchy of knowledge bases. The hierarchy of knowledge bases is given in Table 1.

```
component_function(level(2),class(1),name(prz),
    power_control((+,[]),(-,[])),
    pressure_control((+,[prz_heater]),(-,[prz_spray,prz_out])),
    inventory_control((+,[charging]),(-,[let_down])),
    temperature_control((+,[]),(-,[]))).
line_function(level(2),class(1),name(charging),
    temperature_control((+,[]),(-,[])),
    flow_control((+,[charging_pump,charging_valve]),(-,[]))).
```

Fig.3 Typical Form of Function Representation

Table 1

Hierarchy of Knowledge Bases for Structure and Function

level		class 1	class 2
1	system	RCS	SCS
2	component	Rx, Prz	S/G
2	line	hot leg, cold leg, charging, letdown, Prz Spray, Prz Surge, SI	feedwater, steam line, aux.feedwater, S/G out
3	component	relief tank, RWST	
3	line	Prz out	

3.1.2 Behavior Representation

The behavior of NPP can be represented by using governing equations: mass and energy balance equations.

In some expert systems developed previously, the governing equations of a system are solved numerically using the assumptions given by the generated hypothesis and the calculated results are compared with the real plant data to confirm that hypothesis [11], while the other system attempted to simulate these equations qualitatively

[10]. However, these approaches have some defects. For the previous one, the time required to calculate the consequences of the several hypotheses is too long to do real time diagnosis. Also, the very conditions of most concern in analyzing the behavior of the system under faulted conditions may violate the assumptions under which the simulation models were constructed in the first place. For the second one, the qualitative physics has some limitations to simulate the exact behavior of the plant. Therefore, in HYPOSS, instead of simulating these equations quantitatively or qualitatively, the consistencies of the governing equations are checked using the signals from the plant. This is based on the fact that, as mentioned earlier, many important variables of NPP can be obtained from various instruments or by simple calculations. This is actually similar to the procedure a skilled operator follows in comparing functionally related measurements for consistency with his "mental model" of how the plant behave. Theoretically, this allows us to take full advantage of the known functional relationships between variables when processing the values of their measurements. The consistencies of governing equations of a system are checked with constraints which represent mass or energy balance equation.

The inconsistency of governing equation results the residual which can be used as the measure of inconsistency. According to the amount and relative ratio of residuals, the degree of inconsistency is determined.

3.2 KNOWLEDGE BASE FOR SHALLOW KNOWLEDGE

In HYPOSS, the shallow knowledge is used for three purpose. The first is to represent the characteristics of a specific plant. The second is to represent the facts which are difficult to represent using deep knowledge such as radiation effect. And the third is to diagnose typical or frequently occurring anomalies.

In the first case, the priority can be given to a specific hypothesis among the generated hypotheses using deep knowledge according to the characteristics of a plant. In the second one, the knowledge such as the radiation effect or the complex structural relation which is difficult to represent using deep knowledge are represented using heuristic rules.

The use of shallow knowledge in previous two cases plays the supplementary role for deep knowledge based reasoning. However, the final one is independent of deep knowledge based reasoning. That is, the heuristics part of HYPOSS can be regarded as an independent diagnostic expert system based on shallow knowledge. In other words, through this part only, some typical accidents can be diagnosed which are important or frequently occurred such as Loss of Coolant Accident (LOCA) or reactor

trip. An expert system developed previously in KAIST [18] has been included in HYPOSS as a sub-expert system based on shallow knowledge. Improving this part, the system performance can be improved.

3.3 KNOWLEDGE BASE FOR GUIDELINES

The TMI accident has demonstrated that the guidance provided for mitigating the consequences of design basis accidents could be inadequate when multiple incidents, failures or errors occur during or after the accident. In response to U.S. Nuclear Regulatory Commission (NRC), Westinghouse and the Westinghouse Owners Group have developed new Emergency Response Guidelines (ERGs) [19].

The ERGs are composed of two independent sets of procedures and of a systematic tool to continuously evaluate the plant safety through the response to an accident. The overall organization of ERGs is shown in Fig. 4.

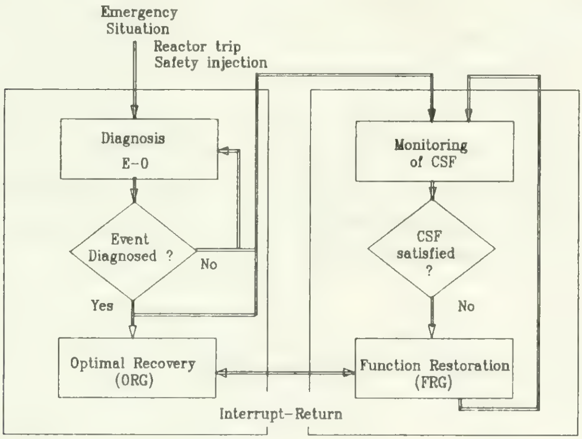


Fig.4 Organization of ERGs

1) Optimal Recovery Guidelines (ORGs)

The ORGs are entered each time the reactor is tripped or the Emergency Core Cooling System is actuated. An immediate verification of the automatic protective actuations is performed and the accident diagnosis process is initiated. When nature of the accident is identified, the operator is transferred to the applicable recovery procedure and sub procedures.

2) Critical Safety Function Monitoring

Early in the course of the accident, the operating staff initiates monitoring of the Critical Safety Functions. These are defined as the set of functions ensuring the integrity of the physical barriers against radioactive release.

3) Function Restoration Guidelines (FRGs)

The FRGs are entered when the Critical Safety Function monitoring identifies a challenge to one of the functions. Depending on the severity of the challenge, the transfer to FRGs can be immediate for a severe challenge or delayed for a minor challenge. Those guidelines are independent of the scenario of the accident, but only based on parameters and equipment availability.

Previous works emphasize the function or symptom based operational aids such as FRGs [2]. However, it should not be considered as a complete alternative to giving the plant operator an early and timely opportunity to revert the plant to normal operating conditions. Also, the need remains for complete diagnosis of anomalies in order to ensure the adequate corrective action is taken. In HYPOSS, the event-based operational guidelines are provided to the operator according to the diagnosed results. If the exact anomalies can not be diagnosed while some of the critical safety functions are challenged, the function-based operational guidelines are provided to the operator. However, the event-based guidelines are provided not only for the anomalies included in ORGs but also for the anomalies which are unanticipated and/or show different behavior with previously analyzed results.

4
INFERENCE STRATEGIES OF HYPOSS

The inference strategy implies how to link the different knowledge representations such as shown in section 3.1 together to diagnosis anomalies. In HYPOSS, at the initial stage of transient, shallow knowledge based reasoning is applied. If the anomalies can not be diagnosed by shallow knowledge based reasoning, then deep knowledge based reasoning is applied. Rule-based deduction and abduction are used for the shallow and deep knowledge based reasoning respectively In this section, only the abduction, i.e. hypotheses generation and evaluation strategies will be presented. Since the shallow knowledge based reasoning process was presented in the previous work [18]. The abduction is based on the human problem solving model in diagnosis tasks have been considered in HYPOSS.

The certainty factor theory developed in MYCIN is adopted as the uncertainty management method [20].

4.1 HYPOTHESES GENERATION

Hypotheses are generated through three steps by forward chaining:

1) overall system state identification step,
2) classification of hypotheses types step,
3) possible cause generation step.

The flow chart of overall hypotheses generation step is given in Fig. 5.

4.1.1 Overall System State Identification

System states are identified to check the change of overall NPP states and maintain an overview perspective. The states of five parameters are identified using rules : power, pressure, inventory, flow rate and temperature. Two systems - reactor coolant system and secondary system (RCS and SCS) are considered in HYPOSS.

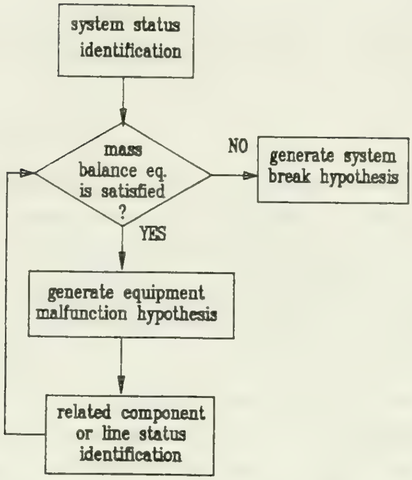


Fig.5 Flow Chart for Hypotheses Generation Step

4.1.2 Classification of Hypotheses Types

In HYPOSS, the generated hypotheses can be divided into two types: equipment malfunction and system break. The equipment malfunction means that the equipment state is different from the proper state in the present situation of NPP. The system break includes the rupture of a line and the leak from a component. They are classified by checking the consistency of the mass balance equation of a system.

If the mass balance equation of a system is satisfied, the type of hypotheses is assumed as the equipment malfunction. Otherwise, the type of hypotheses is the system break. The consistency of mass balance equation is checked during several time steps to mitigate the initial or time delay effect of transients.

4.1.3 Possible Cause Generation

If the type of the hypotheses is the equipment malfunction, then the states of the components, lines and equipments related to the system are identified. The related equipments which can cause the identified system states become the hypotheses directly. Among the related components and lines, the ones whose states can cause the identified system states are selected as the candidates for further hypotheses generation. For the selected candidates, the type classification and cause generation steps are applied

recursively until the related equipments which can cause the identified component or line state are found. Then, these equipments become the hypotheses. This procedure are based on the hierarchy of knowledge bases for structure and function.

If the type of the hypothesis is the system break, then the system or component the mass balance equation of which is not consistent with the signals from NPP becomes the hypothesis. And some parameters such as flow rate through broken area are estimated in this step.

4.2 HYPOTHESES EVALUATION

The hypotheses evaluation is performed through three steps by backward chaining:

1) evaluation using signals of equipment state,
2) evaluation using system behavior,
3) evaluation using shallow knowledge.

4.2.1 Evaluation Using Signals of Equipment State

This step is applied only for the hypotheses of the equipment malfunction . If the signal of a equipment state is consistent with a generated hypothesis, then certainty factor of the hypothesis is increased. The automatic protective action is verified in this step.

4.2.2 Evaluation Using System Behavior

The system behavior can be represented using governing equations. As discussed earlier, the consistencies of the governing equations are checked using the signals from NPP instead of simulating these equations quantitatively or qualitatively. In this step, only the consistency of energy balance equation is checked since mass balance equation is already checked in the hypotheses generation procedure.

4.2.3 Evaluation Using Shallow Knowledge

This step is performed using the shallow knowledge which is explained in the previous section. The efficiency of diagnosis can be improved by this step.

5
VALIDATION STUDIES

The validation studies are performed for several cases. The data for validation studies are produced by a simulator developed in KAIST [21]. The validation of an expert system is to be performed for intermediate results, the final results, the reasoning of a system, or any combination of these three.

Here, small LOCAs due to pressurizer PORV stuck open like in the TMI-2 accident and due to vessel leak of pressurizer are diagnosed as the examples for anticipated and unanticipated anomalies respectively. The validation studies of present work are focused on the performance of deep knowledge based reasoning since the results of the shallow knowledge based reasoning were shown in [18].

The validation studies are based on the assumptions that the signals from NPP are validated and multiple failures are not occurred simultaneously.

In the present stage, the data from the simulator to be diagnosed were prepared in an input file. This procedure is intended to simulate the on-line data transfer from NPP to HYPOSS.

5.1 SMALL LOCA DUE TO THE STUCK OPEN OF PRESSURIZER PORV

The event scenario of this validation study is shown in Table 2 and the diagnosed results are shown in Figs. 6-8.

The identified states of RCS and SCS are shown in Fig. 6. The mass balances of both systems are maintained, so the equipment malfunction is assumed. Then the states of lines related to RCS are identified and the surge line is selected as a candidate for further hypothesis generation since only the flow rate of surge line increases. The surge line has no equipment, so the change of the state of surge line can be caused only from the change of pressurizer. The identified states of pressurizer are shown in Fig. 7. The mass balance of pressurizer is also maintained so the equipment malfunction is assumed again. Finally, the results of diagnosis are shown in Fig. 8. As shown in Fig. 8, three hypotheses are generated : the down of pressurizer heater, pressurizer safety valve open

and PORV open. The certainty factor values show that PORV open is the most probable cause of transient.

Table 2

Event Scenario of Pressurizer PORV Stuck Open

TIME(SEC)	EVENT SCENARIO
5	pressurizer PORV stuck open with capacity 100%
8	reactor trip by pressurizer low pressure signal
18	HPI actuation

```
system_status(time(8),name(rcs),
    status((power,100.66,0.118,steady,0.992188,)
        (pressure,2183.25,-55.109,decrease,0.984375,)
        (inventory,247724,-144.434,decrease,0.984375,)
        (flow_rate,9419.95,-31.556,steady,0.992188,)
        (temperature,606.421,-0.14,steady,0.992188))).
system_status(time(8),name(scs),
    status((power,0,0,steady,0.992188,)
        (pressure,806.922,0.115,steady,0.992188,)
        (inventory,124413,-0.243,steady,0.992188,)
        (flow_rate,0,0,steady,0.992188,)
        (temperature,519.472,0.017,steady,0.992188))).
mass_balance(time(8),name(rcs),cf(0.0347098),check_no(7))).
mass_balance(time(8),name(scs),cf(0),check_no(7))).
```

Fig.6 Identified System Status (I)

```
component_status(time(8),name(prz),
    status((power,0,0,steady,0.5,)
        (pressure,2179.87,-55.554,decrease,0,)
        (inventory,22430.7,3.056,steady,0.5,)
        (temperature,541.366,-0.215,steady,0.5))).
mass_balance(time(8),name(prz),cf(0.843374),check_no(2))).
```

Fig.7 Identified Pressurizer Status

diagnosed_result(8,prz,pressure,decrease,prz_heater,down,0.5).
diagnosed_result(8,prz_out,flow_rate,increase,prz_porv,open,0.875).
diagnosed_result(8,prz_out,flow_rate,increase,prz_sv,open,0.5).

Fig.8 Diagnosed Results (I)

5.2 SMALL LOCA DUE TO THE VESSEL LEAK OF PRESSURIZER

In this study, the data of the previous study are modified and used, since the data for this case can not be produced directly by the simulator. Here, it is assumed that the PORV is closed, i.e. the flow rate through PORV is zero and the loss of coolant by PORV open in the previous case is due to the vessel leak of pressurizer. The diagnosed results are shown in Figs. 9 and 10. The diagnosis procedures are same with the previous case until the states of pressurizer are identified. In this case, the mass balance of pressurizer is not satisfied. So the break of it becomes a hypothesis. The result of evaluation step, as shown in Fig.10, confirms this hypothesis.

```
system_status(time(13,name(rcs),
    status((power,9.213,-91.329,decrease,0.999512,)
        (pressure,1963.15,-275.209,decrease,0.999512,)
        (inventory,249129,1260.77,increase,0.998047,)
        (flow_rate,9467,15.495,steady,0.999023,)
        (temperature,597.809,-8.752,steady,0.999756))).
system_status(time(13,name(scs),
    status((power,0,0,steady,0.999756,)
        (pressure,868.575,61.768,increase,0.999512,)
        (inventory,123859,-553.958,increase,0.998047,)
        (flow_rate,0,0,steady,0.999756,)
        (temperature,528.098,8.642,steady,0.999756))).
mass_balance(time(13),name(rcs),cf(0.518671),check_no(12))).
mass_balance(time(13),name(scs),cf(0.497312),check_no(12))).
mass_balance(time(13),name(prz),cf(0.913642),check_no(7))).
```

Fig.9 Identified System Status (II)

diagnosed_result(13,prz,break,0.97841).

Fig.10 Diagnosed Results (II)

6
CONCLUSIONS

An expert system, HYPOSS, is developed to diagnose anomalies of NPP and to offer the correct operational response guidelines. This system adopts the hybrid knowledge approach to couple the merits of the shallow and deep knowledge approaches. The results of validation studies show that the developed system can diagnose all causes of anomalies if the structure and function description are adequate.

The merits of the hybrid knowledge approach can be found largely in four aspects.

1) The use of structural and functional knowledge provides closure and completeness. Closure is gained by using a simple uniform inference mechanisms to derive a large number of possible faults directly from the description of the system instead of writing hundreds of rules. Completeness is derived from examining all structure connections and paths to generating that nothing is forgotten.

2) HYPOSS estimates and adjusts some parameters based on the signals from NPP to satisfy the consistencies of governing equations. So, the same hypothesis which shows different behavior such as LOCA with different break sizes can be managed in HYPOSS.

3) Using shallow knowledge, the knowledge such as radiation effect, complex structural relation or the special characteristics of a specific plant can be represented and used in deep knowledge based reasoning to improve diagnosis efficiency. Also, the typical or frequently occurring anomalies are diagnosed effectively by an independent expert system based on shallow knowledge.

4) Separation of deep knowledge types provides the high degree of flexibility of modification. Once the structure of NPP is changed, then only the modification of the knowledge bases for structure and function is required to reconstruct HYPOSS.

As the next stage of development, the knowledge base for operational guidance will be extended to give practical operation guidelines. The improvement of consistency check method is also required to treat system behavior more precisely.

REFERENCES

1. A.B.Long, "Computerized Operator Decision Aids," Nuclear Safety, Vol.25, No.4, July-August (1984).
2. D.G.Cain, "Review of Trends in Computerized Systems for Operator Support," Nuclear Safety, Vol.27, No.4, October-December (1986).
3. W.C.Elim, "How Artificial Intelligence Can Help," Nuclear Engineering International, May (1988).
4. W.R.Nelson, "Reactor: An Expert System for Diagnosis and Treatment of Nuclear Reactor Accidents," Proceedings of the 2nd National Conference on Artificial Intelligence, PA, USA, August (1982).
5. N.Yamada et al., "A Diagnosis Method of Dynamic System Using The Knowledge on System Description," Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe (1983).
6. W.R.Nelson, "Response Trees and Expert Systems for Nuclear Reactor Operations," NUREG/CR--3631, February (1984).
7. T.Kiguchi et al., "A Knowledge Based System for Plant Diagnosis," 1985 ANS Int'l Topical Meeting on Computer Applications for NPP Operation and Control, WA, USA, September (1985).
8. T.Washio et al., "Semantic Network Approach to Automated Failure Diagnosis in NPP," 1985 ANS Int'l Topical Meeting on Computer Applications for NPP Operation and Control, WA, USA, September (1985).
9. DE Sebo et al., "RSAS: A Reactor Safety Assessment Expert System," 1985 ANS Int'l Topical Meeting on Computer Applications for NPP Operation and Control, WA, USA, September (1985).
10. M.R.Herbert, G.H.Williams, "An Initial Evaluation of the Detection and Diagnosis of Power Plant Faults Using A Deep Knowledge Representation of Physical Behavior," Expert Systems, Vol.4,No.2, May (1987).
11. J.A.Hassberger et al., "Intelligent Simulations for On-line Transient Analysis," Proceedings of Int'l Topical Meeting on Adv. in Reactor Phys., Math. and Computations, Paris (1987).

12. S.B.Guarro, "A Logic Flowgraph-Based Concept for Decision Support and Management of Nuclear Power Operation," Reliability Engineering and System Safety, Vol. 22 (1988).

13. S.Kaplan, "Outline of COPILOT, An Expert System for Reactor Operational Assistance Using a Bayesian Diagnostic Module," Reliability Engineering and System Safety, Vol. 22 (1988).

14. R.C.Erdmann, Bill K-H.Sun, "An Expert System Approach for Safety Diagnosis," Nuclear Technology, Vol.82, August (1988).

15. R.Milne, "Strategies for Diagnosis," IEEE Trans. on System, Man and Cybernatics, Vol SMC-17 No.3, May-June (1987).

16. P.K.Fink et al., "Expert Systems and Diagnostic Expertise in the Mechanical and Electrical Domains," IEEE Trans. on System, Man and Cybernatics, Vol SMC-17 No.3, May-June (1987).

17. The SD-Prolog Programmers Reference Manual, System Designers Plc, Jan. (1987).

18 K.Y.Choi et al., "The Manipulation of Time-Varying Dynamic Variables Using the Rule Modification Method and Performance Index in NPP Accident Diagnostic Expert Systems," IEEE Trans. on Nuclear Science, October (1988).

19. J.P.Dekens et al., "The Emergency Response Guidelines for the Westinghouse Pressurized Water Reactor," Proceeding of Seminar on Diagnosis and Response to Abnormal Occurrence at NPP, IAEA, Vienna (1985).

20. B.G.Buchanan et al., Rule-Based Expert Systems, Addison-Wesley Publishing Company (1984).

21 Development of Training Simulator Using Personal Computer, KRC-87N-JO2 Project Report (1988).

Preliminary Design of a High-Voltage Power Network: Further Developments of the Expert System Prototype Transept

F. D. GALIANA and J. P. BERNARD
Electrical Engineering Department
McGill University
3480 University St.
Montreal, P.Q., Canada H3A 2A7

D. MCGILLIS and C. KRISHNAYYA
Hydro-Quebec
Montreal, P.Q., Canada J4S 1M5

ABSTRACT

This paper describes an expert system prototype for the preliminary design of an AC/DC transmission network. Several aspects of this process are presented: Motivation and justification, objectives and benefits, interaction between knowledge engineer and domain experts, characteristics of the prototype, examples of expert system designs.

INTRODUCTION

The power industry in many parts of the world has, over the past ten years, experienced a very slow growth, and many utilities are concerned that the vast body of power system design expertise acquired by its engineers during the decades of the 60's and 70's will gradually erode with retirements and inactivity. Companies such as Hydro-Québec are therefore looking at expert system technology as a potential approach to represent and preserve some of the expertise and knowledge presently available in the minds of human experts.

The area of power system design is one where the loss of expertise is strongly felt, particularly during periods of slow growth. There are many decision levels of the power system design process where

"human expertise" from a variety of disciplines plays an important role. Among these we can mention the selection of: generation type, generation site, transmission type (AC/DC), voltage level, number and type of transmission lines, number and type of compensation devices. These decisions must be weighed with respect to several criteria including cost, environmental aspects, reliability, stability, and steady-state performance. Human experts are aided in the design process by digital simulation tools such as load flow, transient stability, EMTP, and analog tools such as the transient network analyzer and the HVDC simulator which allow the designer to examine the fine details of the design and make minor adjustments. It is interesting to observe, however, that the first pass at a design by a human expert is usually based on simple models or on heuristic rules derived from practical experience.

This paper describes an expert system prototype for the preliminary design of an electric transmission network. The objectives and expected benefits of such a prototype are also discussed.

OBJECTIVES AND EXPECTED BENEFITS OF THE ES

The principal objective of this project was to investigate the potential of expert systems (ES) to simulate the behaviour of human experts in the preliminary design of AC/DC power transmission networks. This implied identifying and representing in a systematic manner the knowledge needed to arrive at an acceptable design or designs. This body of knowledge includes symbolic and numeric data, as well as

principles, rules, guidelines, trade-offs and mathematical models governing such designs. The acceptable designs are based on cost, environmental aspects, stability, reliability and design sensitivity or robustness. The prototype ES should also allow the user to interact with this knowledge base by guiding him or her through the various steps required to arrive at a design, as well as providing the user with an explanation of the line of reasoning followed to arrive at each of the intermediate steps.

The principal benefits of the study are:

(1) Expert knowledge can be retained and passed on to future engineers,
(2) Knowledge can be expressed in a more concrete form, rather than on unexplicit or unfounded decisions,
(3) More consistent results will be obtained,
(4) The ES will be able to examine more cases than a human expert, potentially leading to better and more economical designs,
(5) The ES can act as a teaching and training tool for future experts through its reasoning explanation facility,
(6) The ES can be helpful to experts as a "consultant" or as a means to evaluate or improve a given design,
(7) The ES will act as a precursor to a family of expert systems used for the design of different components of a power system, acting as a model from which a more general design methodology could be extracted,
(8) The ES serves to centralize or focus knowledge which often is dispersed throughout an organization.

DOMAIN EXPERTS AND KNOWLEDGE ENGINEER INTERACTION

In this project the knowledge engineers had experience in both the domain (power system analysis and design) and in the AI area, a mixture which proved very useful in extracting and condensing expert knowledge from the domain experts. The latter were power system planning engineers with many years experience in the planning, design, and debugging of major projects such as James Bay in Canada and Itaipu in Brazil.

POINT-TO-POINT TRANSMISSION SYSTEM DESIGN

The prototype ES considers the preliminary design of an AC/DC transmission network carrying power from a distant generation source, usually hydro, to the load centers. For the purpose of a preliminary design it is assumed that the network is point-to-point and that the voltage and frequency at the receiving end are constant. This and other assumptions will be removed at a later date for more advanced prototypes. The AC network may contain several parallel lines, static var compensators, series capacitors, shunt reactors and intermediate substations. The DC transmission network being point-to-point, will not have intermediate substations. It may contain more than one 12-pulse valve group per pole, several parallel bipolar lines, various combinations of AC filters and other reactive power supply equipment at the terminal substations, different types of DC filters on the lines and various operating conditions.

EVALUATING A DESIGN

A candidate design must meet the following conditions:

- The minimum fault recovery characteristics of the DC system.
- Permissible radio and harmonic interference requirements.
- The minimum stability criteria following a fault.
- The steady-state voltage and reactive generation limits.
- The temporary overvoltage limits following load rejection.
- The minimum equipment redundancy imposed by reliability.
- It must be insensitive to input data variations (robust).
- It must meet some maximum cost requirements
- It must fall within the available state-of-the-art technology.

EXPERT DESIGN METHODOLOGY OF A POINT-TO-POINT TRANSMISSION NETWORK

Expert designers of power transmission networks are engineers with extensive knowledge of actual designs and their operational performance, as well as of the scientific and mathematical principles governing their behaviour. Their knowledge includes a combination of empirical rules, mathematical relations and facts. This knowledge allows them to narrow down the search for a reasonable design to a small subset of potential candidates which are then analyzed and refined with elaborate mathematical simulations. The results of the simulations in turn enrich their knowledge base and their expertise.

The main design methodology described by experts in power transmission (McGillis, Krishnayya, Hotte, Chahine, Peixoto, Gyugyi) is as follows (there exist individual preferences as to the order in which the various steps are executed, but there is agreement as to the approach):

1. Read transmitted power and line length
2. Recommend AC or DC transmission

AC TRANSMISSION

3. Recommend a trial AC voltage level.
4. Recommend a trial number of circuits.
5. Recommend a trial number of intermediate substations.
6. Recommend sufficient shunt reactance to keep the load rejection temporary overvoltage to less than 1.4 p.u. Add an equal percentage of series capacitance in order to maintain var balance in the line during heavy load.
7. Recommend a minimum level of series and/or shunt compensation needed to maintain stability during heavy load.
8. Recommend minimum reactive var injection of static var compensators (svc) during light load conditions to absorb excessive vars. Alternatively, recommend additional switchable shunt reactors.
9. Calculate costs of lines, transmission losses, shunt reactors, svc's, series compensation, and substations.
10. Repeat design process for several values of nominal voltage, number of circuits, number of intermediate substations, and compensation

strategy around trial values. Test flexibility of design to escalating costs. Recommend one or more candidate designs according to lowest cost and/or low sensitivity to escalating costs.

DC TRANSMISSION

3. Recommend a trial DC voltage level.
4. Recommend a trial number of DC bipolar lines.
5. Recommend a trial number of valve groups per pole and their type.
6. Recommend the mix of VAR compensation equipment connected to the interconnecting AC bus.
7. Recommend type and size of AC and DC filters and smoothing reactors.
8. Recommend type, size and arrangement of DC arresters.
9. Calculate cost of converter station, station and line losses, and DC line.
10. Repeat design process for alternative values of DC voltage, number of valve groups, AC interconnection voltage, and VAR compensation equipment. Test the design flexibility to expanding costs. Recommend one or more designs according to the lowest cost and/or low sensitivity to escalating costs.

KNOWLEDGE CHARACTERIZATION

Expert design knowledge is represented using the concepts of assertions, frames, and rules. The inference engine executes in a forward chaining mode utilizing a number of special features such as frame-inheritance, rule-sponsors, daemons, rule priorities, and rule

dependency. Up to now about 50 basic rules have been identified which lead to a set of acceptable point to-point transmission network designs.

DESCRIPTION OF EXPERT SYSTEM PROTOTYPE

The expert system prototype operates on an IBM PS/2 Model 80 with a VGA color screen and 10 Mb of Extended RAM. It is based on the expert system development shell GoldWorks. The ES and the man-machine interface are based on Common Lisp and on a power system simulator which is written in Microsoft QuickBasic. The user interacts with a menu-driven interface, a mouse and a keyboard. The principal features of the ES are:

. The user can examine and modify the type and characteristics of the equipment available for design by the ES.
. The user can run and trace the inference engine.
. The user can examine in summary form or in complete detail the designs produced by the ES.
. The user can request a description of the reasoning chain that led to the assertion of any fact or data derived by the ES.
. The user can fine-tune any ES derived design and test its performance through the simulator.

The inference engine of the expert system is run whenever the user enters, as a minimum, the maximum power transmitted and the distance between generation and load. Design rules are executed by the inference engine, some of which specify design parameters, while other rules run simulations whose results may fire new rules which eventually

generate a set of acceptable designs. The ES architecture in block diagram is shown in Figure 1.

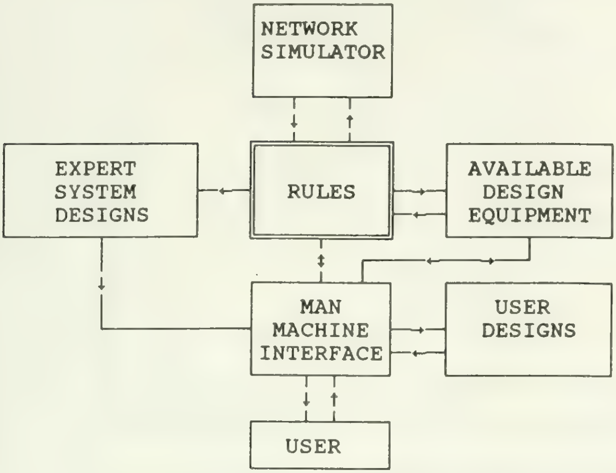


Figure 1: Expert System Architecture

EXAMPLES OF EXPERT SYSTEM OPERATION

Figures 2 through 8 indicate a sample of the menus displayed by the ES under different conditions of the AC transmission case. Positioning the mouse controlled cursor on any title causes a new menu to pop up on the screen offering a new set of options, or some new results, or an explanation of a derived fact. The man-machine interface is very friendly, requiring a minimum of user input in the form of text or data. Figure 2 shows two typical popup menus, one indicates the types of data

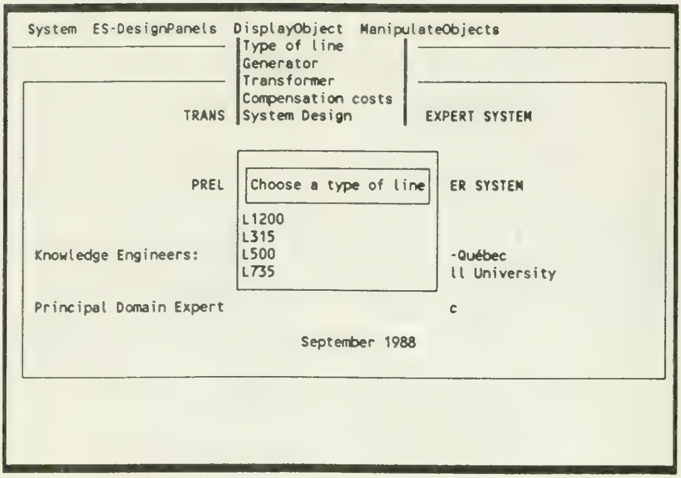


Figure 2: Background Data and Line Type Menus

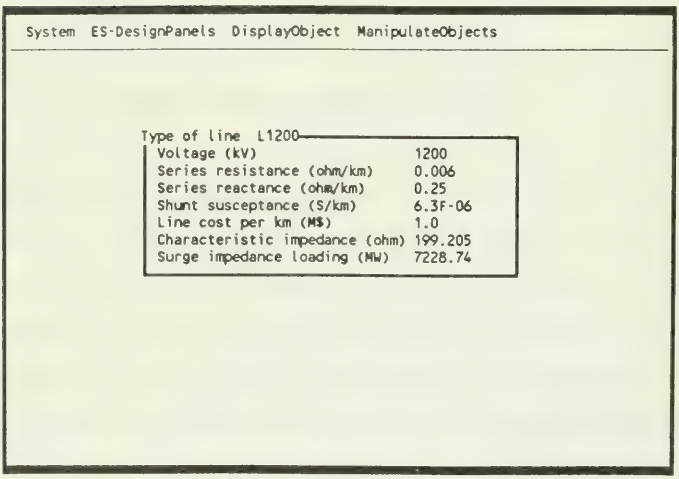


Figure 3: Line Parameters for Line Type L1200

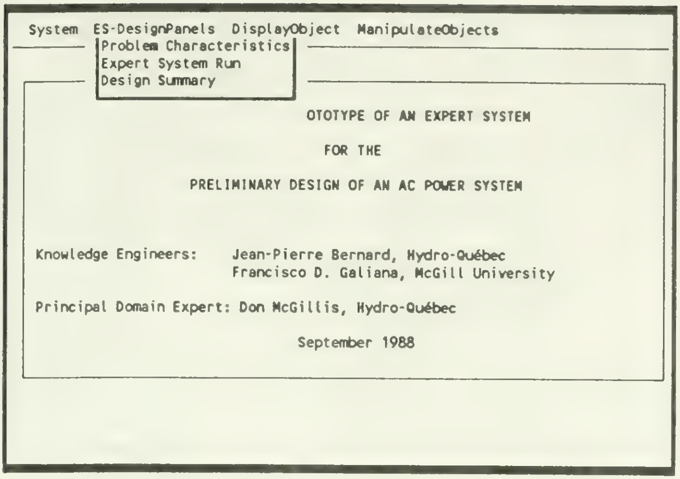
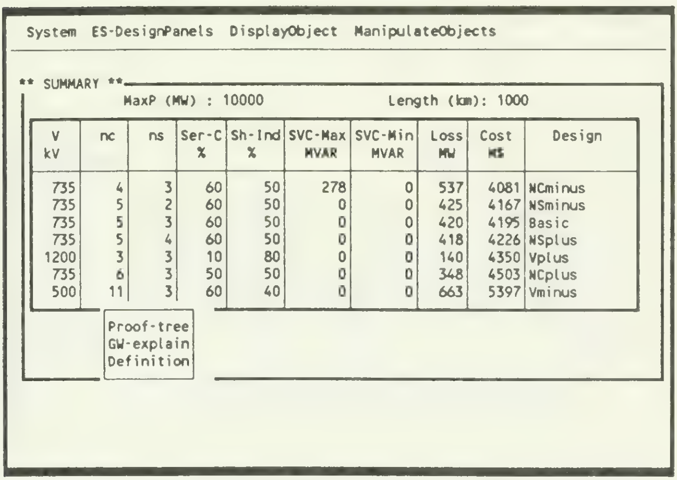


Figure 4: Design Options Menu



V kV	nc	ns	Ser-C %	Sh-Ind %	SVC-Max MVAR	SVC-Min MVAR	Loss MW	Cost M$	Design
735	4	3	60	50	278	0	537	4081	NCminus
735	5	2	60	50	0	0	425	4167	NSminus
735	5	3	60	50	0	0	420	4195	Basic
735	5	4	60	50	0	0	418	4226	NSplus
1200	3	3	10	80	0	0	140	4350	Vplus
735	6	3	50	50	0	0	348	4503	NCplus
500	11	3	60	40	0	0	663	5397	Vminus

Figure 5: Summary of ES Designs Menu

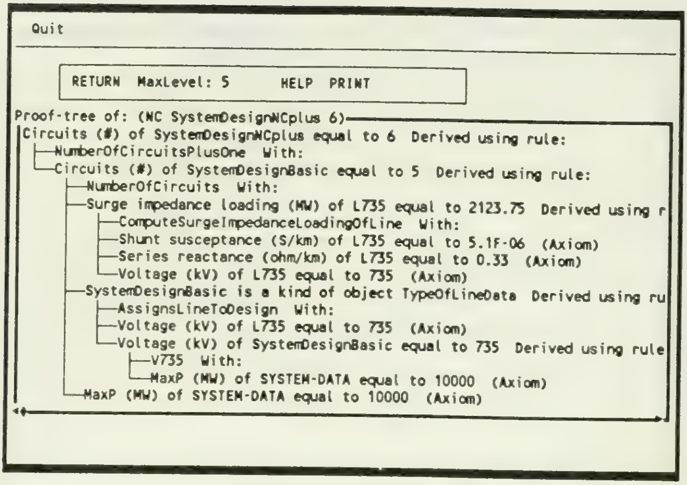


Figure 6: Reasoning Logic of ES Menu

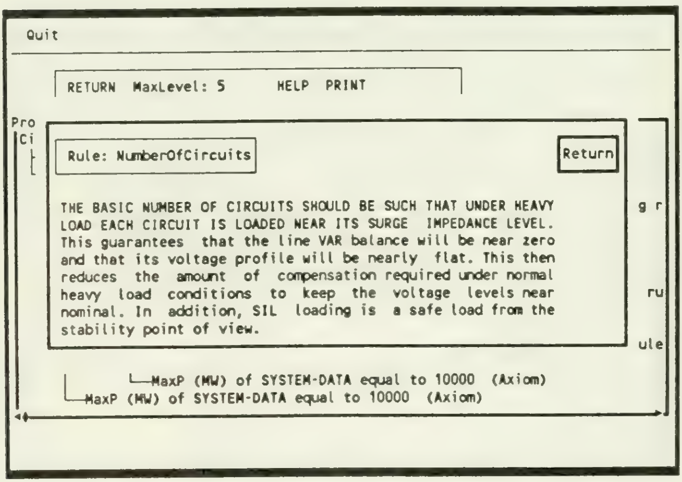


Figure 7: Rule Explanation Menu

```
System  ES-DesignPanels  DisplayObject  ManipulateObjects

Transmission System Characteristics
  MaxP (MW) 10000    MinP (MW) 5000     Length (km) 1000   MaxQ (MVAR) 5000
  MinQ (MVAR) -1000     Fault duration (cycles) 6
  Sect.out before fault (#) 0

System Design SystemDesignVplus
  Voltage (kV) 1200     Circuits (#) 3   Substations (#) 3
  Series cap.(%) 10     Shunt ind.(%) 80     Max. SVC (MVAR) 0.0
  Min.SVC (MVAR) 0.0        Stable? YES  Max.TOV (pu) 1.4   TOV (pu) 1.27232
  Losses (MW) 139.59    Cost of system design (M$) 4349.94

More Data                   Computed costs
Objects used                  Cost of line (M$)            3000.0
 -Line                        Cost of series cap.(M$)      40.9868
 -Generator                   Cost of shunt ind.(M$)       702.502
 -Transformer                 Cost of transf.(M$)          59.1429
 -Comp. Costs                 Total cost of gener.(M$)     21000.0
Comp. Levels                  Cost of SVC (M$)             0.0
Powers/Angles                 Cost of losses (M$)          348.988
Voltage Profile               Cost of substations (M$)     198.321
PI equivalent                 Cost of system design (M$)   4349.94
Costs
```

Figure 8: Details of Design Vplus

objects (describing available design equipment), while the center menu shows (upon selection of Type of line) the types of lines available. Figure 3 indicates the menu corresponding the selection of line L1200. Figure 4 shows the principal choices offered by selecting the ES Design Panels menu item. Upon selection of Design Summary one obtains a menu as shown in Figure 5 (after running the ES). This summary contains the main parameters of the designs determined by the ES. By selecting any of the parameter values indicated one obtains the popup menu shown at the bottom of Figure 5. The Definition button gives a short explanation of the meaning and units of the quantity. The Proof-tree button generates the tree illustrated in Figure 6 which justifies the reasoning (and expertise) behind the fact being examined, i.e. the rules used and the facts triggering those rules. By selecting any rule name in the proof tree one obtains a natural language enunciation of the rule and its justification as shown in Figure 7. Finally Figure 8 shows the details of the design costs and other data after selecting the Examine a System Design button, the design Vplus, and clicking the option Costs under the menu More Data.

ACKNOWLEDGEMENTS: We are grateful for the help and encouragement of Mr. R. Manoliu from the Hydro-Québec Task Force on Expert Systems, as well as for the support and facilities provided by the Hydro-Québec Planning Department. The support of NSERC and FCAR is also acknowledged.

REFERENCES

1. Personal communication with Y. Hotte, Hydro-Québec Planning Department, 1988.

2. Personal communication with R. Chahine, Hydro-Québec Planning Department, 1988.

3. Personnal communication with C. Peixoto, Furnas, Brazil, February 1988.

4. Personal communication with L. Gyugyi, Westinghouse R&D Center, Pittsburgh, 1988.

5. Personal communication with R. Naggar, Hydro-Québec, 1988.

6. D.A. Waterman, A Guide to Expert Systems, Addison-Wesley, 1986.

7. "Gold Works Expert System Reference Manual", Version 1.0, Gold Hill Computers, Cambridge, Massachusetts, May 1987.

Methods for Improving the Development and Maintenance of Plant Operating Procedures

CHARLES P. HORNE
CPH, Inc.

ROBERT COLLEY
Electric Power Research Institute
3412 Hillview Avenue
Palo Alto, California 94304, USA

JOHN M. FAHLEY
Pacific Gas and Electric Co.
San Francisco, California 94106, USA

ABSTRACT

This paper describes an ongoing EPRI project for improving the processes of developing, maintaining, and verifying nuclear power plant procedures.

The first task of this project was to evaluate the applicability of structured software analysis methods to operating procedures. It was found that these methods offer significant potential benefits if procedures can be cast into a software-like structure.

The second task is to develop software to compile and verify procedures. The procedure compiler will read and determine procedure structure, and will allow the knowledge expressed in procedures to be accessible to other computer software. The procedure verifier applies specific logic tests to the procedures structure.

The goal of the work is to ease the process of developing and maintaining plant procedures by reducing time and manpower needed to produce quality procedures consistent with plant operation constrains and requirements.

1 INTRODUCTION

Over the past several years there has been considerable progress in software engineering methods for improving the processes of developing and maintaining complex computer software. These methods are still evolving, but have significantly enhanced software quality.

The objective of this project was to develop a plan to adapt software engineering methods to the development and maintenance of plant procedures. The motivation

was the observation that these methods address problems common to both computer software and procedures.

Utilities have made large investments to improve operating procedures. These efforts have concentrated on engineering content and human factor issues. This project focuses instead on managing the complexity of procedures by streamlining the process of procedures development, maintenance, and verification.

In this project the relationship between operator and plant is viewed as analogous to a conversation in which the operator speaks to the plant through control actions, and the plant responds to the operator with observable changes in process state. From this perspective, the objective of this project is to help the operator better communicate with the plant, to correctly perceive the current plant state and to move the plant to the desired state.

The output of this project will be :

- a report outlining the applicability of software engineering methods to the development and maintenance of procedures
- computer software that compiles procedures text according to a prescribed vocabulary and syntax to construct a representation of the procedures that expresses the structure of procedures and the relation of procedures to plant systems
- computer software that applies a set of specific logic tests to the procedures structure to verify their consistency

2 BACKGROUND

From a institutional perspective, operating procedures represent a utility's methods for reaching and maintaining desired plant states. A typical power plant uses thousands of operating procedures, which are intended to cover all likely plant scenarios. At any one time, several procedures can be concurrently active, yet most procedures are inactive. Some procedures are rarely used, others are used daily. Each procedure has an effect on the overall operability of the plant, yet because of the number and complexity of procedures, it is difficult to assure that procedures do not interfere with each other and are valid in the current plant state.

Procedures are written by knowledgeable people, usually following guidelines such as those published by INPO [1] and the NRC [2]. These guidelines are designed to assure that procedures are clearly written and easily performed, and that common plant situations are addressed. Most utilities are adopting standards to promote uniformity among procedures. In some cases, procedures can be accessed via computer.

There are similarities to the current state of procedures development and the early days of software development, when no systematic design process existed and the sole criteria of success of software was that 'it works'. As programs became more complex, development and maintenance costs rose dramatically, major projects failed, and the software engineering discipline emerged to establish principles of good software

design. Software engineering is still evolving and stands today as a collection of systematic approaches and methods [3] covering all phases of software life cycle, such as :

- project planning and management
- requirements analysis
- design
- quality assurance
- testing strategies and techniques
- configuration management

All of the above topics are relevant to procedures.

The objective of requirements analysis is to construct a well formed specification, which describes the information and process structures required to perform a task. Software design transforms the specification into an implementation plan. Design can be further classified by approaches that address different task characteristics :

- data flow oriented design
- real time design
- data structure oriented design
- object oriented design

All these approaches have relevance to procedures, but object oriented design is the most general and best matches the requirements of procedures.

3 EXPECTED BENEFITS

The methods described in this paper are expected to help procedure writers better understand the interrelations of procedures, and the relations of procedures to other plant information. The methods address some of the issues discussed in a NRC review[4] of plant operating procedures. Some of the immediate benefits expected of the project are as follows :

- Complexity control :
  Procedures have complex interdependencies, so assuring the logical consistency of a system of procedures can be difficult. This project will provide tools to manage complexity.

- Enhanced maintainability :
  Since procedures will be archived as computer files, they can be maintained by conventional configuration control software, such as SCCS (Source Code Control System) in UNIX[5], to coerce automatic evaluation of the effects of procedure changes.

- Improved standardization :
  The compilation of procedures will assure conformance to specified vocabulary and syntax, while allowing customization of both vocabulary and syntax. The

vocabulary will support links to other sources of plant information. Procedures documents will be printed in a prescribed format by special software.

In addition to the above immediate benefits, significant indirect benefits are expected. The capability of representing the knowledge in procedures on a computer could lead to several new ways to improve the understanding of the relation of the procedures to plant organization, dynamics, and constraints. Some of the additional benefits might be;

- Tracing procedures to their engineering basis
  Figures 4 and 5 illustrate how the engineering decisions that are input to procedures development process are lost in the procedures document. Access to the reasoning behind a procedure would aid the maintenance process.

- Enhanced on-line procedures guidance :
  An procedures guidance system assures that the operator is aware of all ongoing procedures and the current location within each active procedure. A basic Emergency Operating Procedures guidance system was developed by EPRI in collaboration with TaiPower (6). An on-line procedures system can give more detail and present explanations.

  Future procedures guidance systems will incorporate massive support information. Even now a single 4" CD-ROM disk can store about 11 feet of equivalent paper documentation, which could probably store all plant procedures, drawings, FSAR's, fault trees, and illustrations.

- Automated procedures generation :
  A on-line procedure generator extends the procedures guidance concept to synthesize procedures on-line, based on current plant conditions. A prototype procedure generator system has been demonstrated by Colley (7). An on-line procedures generator can provide procedures support for plant conditions deemed too rare to include in a static printed document. Procedure documents are written to be data driven for simplicity, but the process is inherently goal driven: an on-line procedure generator can pursue goals and still present simple data driven instructions.

- Generality :
  Although operating procedures are the immediate focus of this project, the methods are applicable to all types of procedures.

- Improved simulations :
  The representation of procedures structure will enable plant simulations to incorporate effects of operator actions under various levels of procedure compliance. These studies could improve estimates of performance and risk.

- Procedure optimization :
  In cases where more than one procedure can satisfy a goal, a software program could determine the best procedure based on any reasonable figure of merit based on time and complexity measures.

- Isolation :
  The isolation boundaries required for the performance of some maintenance procedures could be automatically determined, since the plant components affected by each procedure would be available to software.

4 APPROACH

Two stages to improving development and maintenance of procedures have been identified : (1) a compile and test approach and (2) an integrated workstation approach. These stages are analogous to a software language compiler and a computer aided software engineering (CASE) workstation : compilers have been in use for years, and CASE workstations are just beginning to be used.

In the compile and test approach, a set of tests are applied to procedures. This approach is envisaged as illustrated in figure 1.

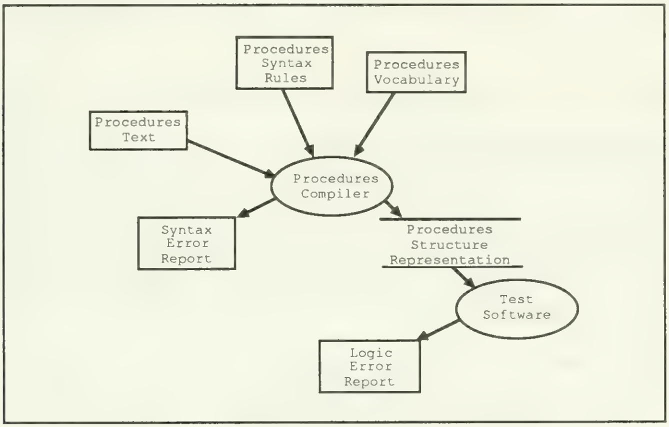


Figure 1. The Compile and Test Approach

Experience with complex software has shown that testing alone does not guarantee quality, that quality begins at the conceptual stage of software development. The integrated workstation approach seeks to implicitly assure quality by integrating analytic tools and information required for procedure development and maintenance. This approach incorporates the compile and test approach.

The plan for this project is to implement the compile and test approach, and some features of the integrated workstation approach.

5 CONTEXT

This section describes a high level view of the context of procedures in the utility. Procedures are part of the information processing function of the illustration in figure 2.

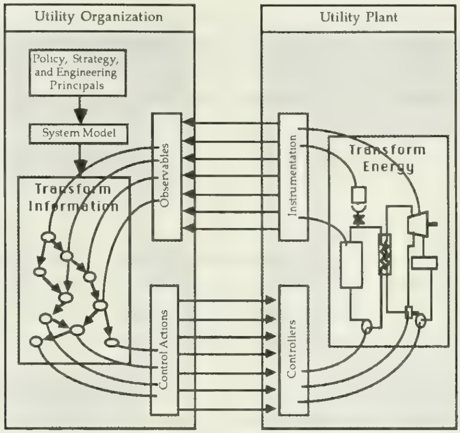


Figure 2. The High Level Relations Between a Utility, Information, and Energy

The plant information elements related to procedures include :

- the connections between plant components :
  This incorporates the information in engineering drawings.

- the states of plant components and systems
  States of simple components may be described by discrete values of control variables or continuous values of process variables. States of complex systems are described by expressions incorporating states of components.

- constraints on component and system states :
  There are several types of constraints - physics constraints from basic conservation laws, economic constraints, and safety constraints.

- a set of goals :
  These are expressed as critical safety functions, or as the titles of normal operating procedures.

This project focuses on logical connections of procedures, including interprocedural connections and connections between procedures and plant systems, processes, and constraints. Some of the types of connections are illustrated figure 3.

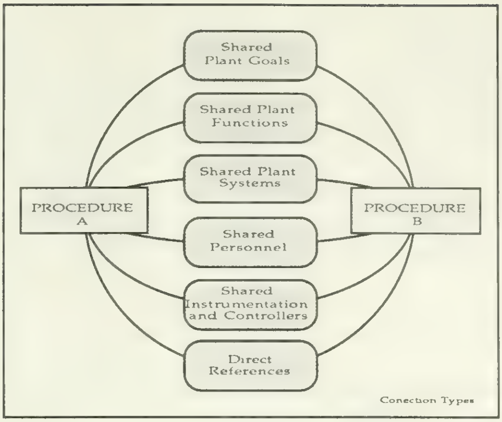


Figure 3. Some of the types of connections between procedures

6 THE PROCEDURES DEVELOPMENT PROCESS

Procedures development is guided by a model based on engineering knowledge and experience, as illustrated in figure 4. For each anticipated plant state and each desired plant state, the model is used to predict suitable rules that will move the plant from the current state to the desired state. The states used for rule premises depend mostly on process variable such as temperature and pressure, and the rule actions adjust alignment variables (the states of control elements such as pumps and valves) to cause the plant process state to evolve towards the desired state.

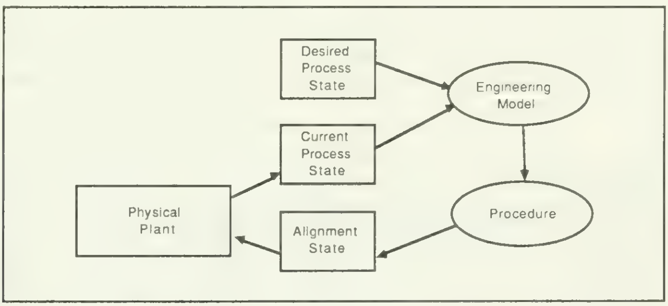


Figure 4. Information Flow in Procedures Development

Later, when the procedures are being used, the model is replaced with the rules of the procedure, as illustrated in figure 5. The engineering reasoning underlying the procedure is inaccessible at execution time.

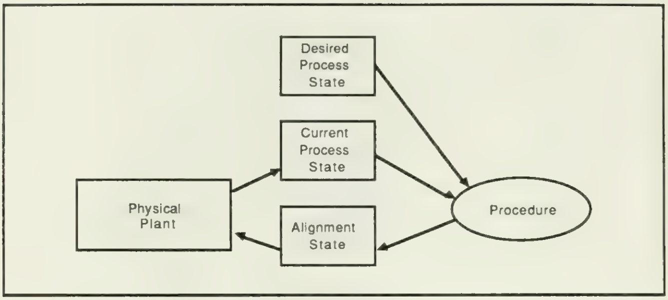


Figure 5. Information Flow in Procedures Delivery

Compared to rules used in most expert systems, the procedures are written to be data driven with little true inferencing. However, the process of writing procedures is inherently goal driven.

7 PROCEDURES COMPILATION

This section describes how computer software can read the procedure text to recognize and represent the procedure structure, and how procedures can be automatically related to other sources of plant information.

To understand text documents, the computer software must relate words from the text to a model defined by syntax rules and vocabulary. Procedures such as those developed by the Westinghouse Owner's Group [8] have the structure depicted by the abbreviated syntax rules shown in figure 6.

Procedure	←	BoilerPlate Step+
BoilerPlate	←	title scope category date revision symptoms notes
cautions conditions		
Step	←	Action ExpectedResponse ResponseNotObtained
Action	←	Text Instruction+
ExpectedResponse	←	Text Instruction+
ResponseNotObtained	←	Instruction+
Instruction	←	Rule | Imperative | Note | Caution
Rule	←	ifExpr thenExpr thenAction

Figure 6. The Syntactic Elements of Westinghouse Procedures

In this notation, based on the Bacus-Nauer form[9] of production rules, the trailing + means 'one or more', the | means 'or', and items separated by spaces are related by 'and's. Thus for example, <u>ExpectedResponse</u> is composed of <u>Text</u> followed by one or more instances of <u>Instruction</u>, and each instance of <u>Instruction</u> is composed of either a <u>Rule</u> or a <u>Imperative</u> or a <u>Note</u> or a <u>Caution</u>.

A sample of procedure text is shown in figure 7, slightly augmented by labeling the steps and rules. Each Step has three sections, the Action, ExpectedResponse and ResponseNotObtained. The Action section specifies an action that is always performed. The optional ExpectedResponse section specifies the response normally expected. Note that in Step 2 of the example the ExpectedResponse is implied as part of the Action. If the expected response is not obtained, then the ResponseNotObtained section is activated.

```
Step 1. ACTION :

Check PZR level and charging flow

Step 1. EXPECTED RESPONSE :

No significant changes in PZR level OR charging Header Flow.

Step 1. RESPONSE NOT OBTAINED :

Rule 1.1 :
IF  PZR level is dropping OR charging header flow is increasing,
THEN START additional charging pump.

Rule 1.2 :
IF  level continues to drop,
THEN ISOLATE letdown:
 o  CLOSE letdown orifice valves CVCS-8149A, B, C
 o  CLOSE letdown isolation valves LCV-459/460

Rule 1.3 :
IF  level is still dropping
THEN manually initiate SI and GO to EP E-O, "REACTOR TRIP OR
            SAFETY INJECTION".
```

Figure 7. Example Procedures Text

The vocabulary input to the compiler will be a list of words together with their possible syntactical roles. Alignment and process state variables will have the form Variable ← PlantComponent VariableName VariableValue.
For example, in figure 7, the expression PZR level is dropping appears: PZR is a PlantComponent, level is a VariableName, and is dropping is a VariableValue. There are many technical problems in the recognition process. Later in the same example, PZR level is referred to as simply level. Also later the VariableValue becomes is still dropping.

The internal representation will be object-oriented. Each PlantComponent will be represented as an object having properties referred to in the procedures text.

Information in the procedures can be symbolically related to other sources of plant information. Figure 8 shows some of the major classes of objects in the plant system, such as procedures, plant organization, observables, procedure writer plant model, operator, operator plant model, controllers, and plant.

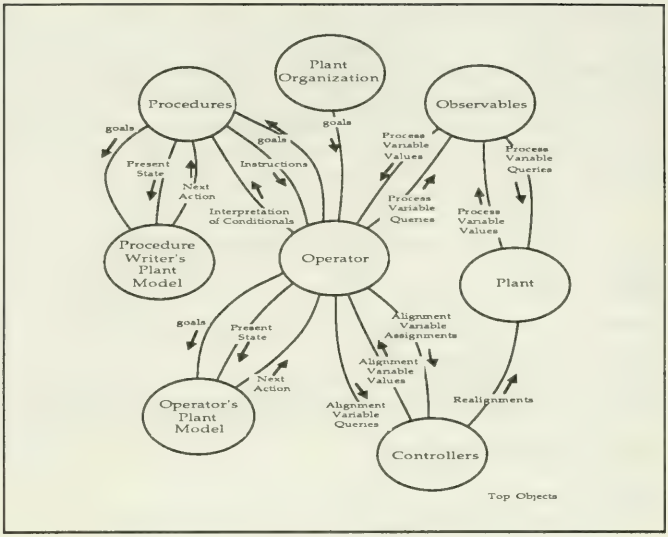


Figure 8. The Principal Objects in the Procedures Model

There may be independent plant models held by various agents. For example, the plant operator's model may differ from the procedure writer's model. Having different models is natural, but communication requires some degree of commonality, acquired through experience and training.

It is an objective of this project that existing text files serve as the archive of procedures, but it may be necessary to annotate existing procedures or add new syntax rules to resolve some difficult natural language recognition problems.

8 QUALITY ASSURANCE TESTS

This section describes some automated tests of procedure structure. The tests can be performed without knowing the physical dynamics of plant processes. Some of the tests are :

- Closure

  The concept of closure is that given a set of states S and a set of state transitions T, S is closed under T if all transitions in T applied to all states in S lead to states in S. This test assures that the transitions induced by procedures lead to acceptable states, as illustrated in figure 9.

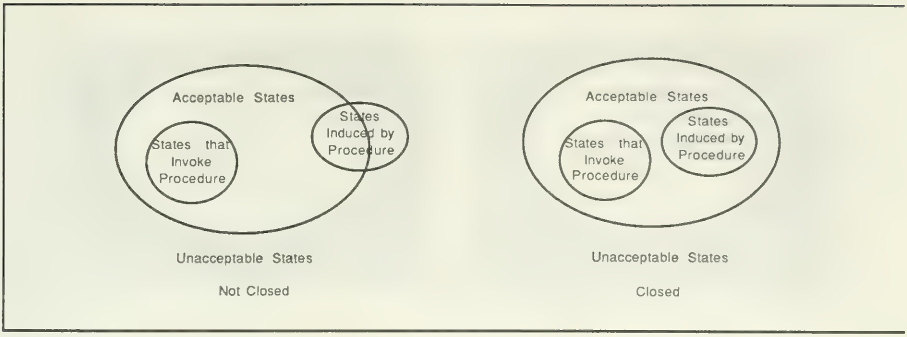


Figure 9. Closed Proceduress

To illustrate this form of closure, consider the system in figure 10 and the procedure rules :

Rule 1 :
IF ((Level 1 < Level 2) AND (Valve 3 IS Closed))
THEN (Turn Pump ON)

Rule 2 :
IF (Pump ON)
THEN ((Open Valve 1 AND Valve 2) AND (Close Valve 3))

The above rules are not closed in the above sense, since if (Valve 1 IS Closed) or (Valve 2 IS Closed) are true, then rule 1 leads to an undesirable state in which (Pump IS on) and the pump could be damaged. A preceding imperative (TURN Pump off) is required.

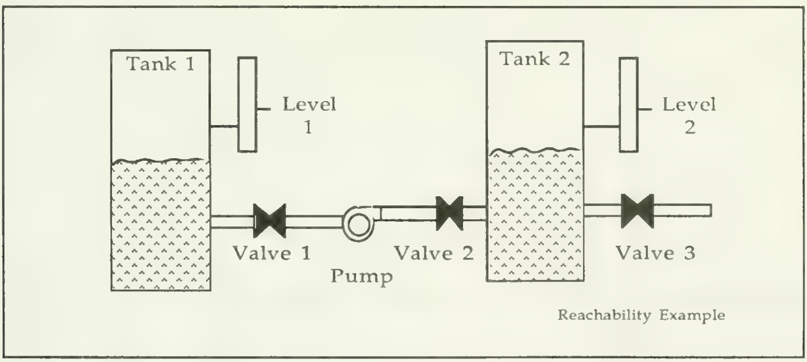


Figure 10. A Simple System for Illustration of Structure Tests

- Proper completion

  Procedures are invoked upon certain entry conditions, but, unlike computer software, may not have a well defined end. The effects of 'dangling procedures' (procedure B initiated from procedure A, procedure A's symptoms disappear, but procedure B remains active) could be evaluated.

- Circular Logic

  The circular logic test searches for rules where the premise of one rule (or step) is the conclusion of another, and deal lock or non-terminating cycle may issue, as in the case of the following rules :

```
Rule 1 :
IF      (Valve 1 IS Open)
THEN    (Open Valve 2)

Rule 2 :
IF      (Valve 2 IS Open)
THEN    (Open Valve 1)
```

  The test program will derive the topological circuits in state space.

- Conflict

  The conflict test searches for competing resources or incompatible states required by two or more concurrent procedures. One way to perform the test is to find all instances of conflicting state variable values, then determine if the conflicting procedure could be invoked concurrently.

- System Component Requirements

  This test checks that the prerequisites of any action are met before proceeding with the action.

  For example, the rules

```
IF      (Level 1 < MinLevel of 1)
THEN    (Open Valve 1)

IF      (Level 2 < MinLevel of 2)
THEN    (Start Pump)
```

  would violate this test because pump could be started without Valve 1 being open.

  This test might fail when GO TO instructions are issued from one procedure to another, bypassing the usual entry conditions of the second procedure.

- Constraints

  Constraints limit the allowed plant states, and come from sources like the FSAR's (Final Safety Analysis Report) and the Technical Specifications

  Complexity of the plant, procedures, and constraints can make it difficult to write procedures that avoid unacceptable states. Although inclusion of comprehensive constraints may be beyond the present scope of this work, the mechanism for testing constraint compliance is planned.

- Merit

  This test provides the ability to find the best procedure among alternatives. The test evaluates a procedure's figure of merit, based on time and complexity measures such as the number of steps to completion, the number of plant states traversed, the number of plant state changes, the proximity to disallowed states, or the minimal expected time to completion.

9 SUMMARY

This research investigates the application of software analysis and design methods to the development, maintenance, and verification of plant procedures. The approach is to analyze the structure of procedure sets, to link the structure to plant states and constraints, and to evaluate logic tests.

The methods and tools developed in this project could yield significant benefits to plant operations.

References

1. Emergency Operating Procedures Writing Guideline, Institute of Nuclear Power Operations, Atlanta, Georgia, June 1982

2. Guidelines for the Preparation of Emergency Operating Procedures , NUREG-0899, U.S. Nuclear Regulatory Commission, Washington, D.C., 1982

3. Robert S. Pressman - Software Engineering - A Practitioner's Approach, McGraw-Hill Book Company, New York, 1987

4. Study of Operating Procedures in Nuclear Power Plants : Practices and Problems , NUREG/CR-3968, U.S. Nuclear Regulatory Commission, Washington, D.C., 1982

5. D.M. Ritchie, K. Thompson, The UNIX Time-Sharing System , Communications of the Association for Computing Machinery, v17, n7, pp365-375 July, 1974

6. W. Petrick, K. Ng, Emergency Operating Procedures Tracking System, NP-5205M, Electric Power Research Institute, June 1987

7. R.W. Colley, J.C. Danley, L.M. McWethy, S.E.Seeman, K.R. Splattstoesser, Procedure Prompting System, HEDL-TC-2242, Hanford Engineering Development Laboratory, August, 1982

8. Emergency Response Guideline, Westinghouse Owner's Group

9. A. Aho and J. Ullman, Principals of Compiler Design, Addison-Wesley Publishing Company, Reading Mass 1977

INTERVIEW[R]: A Program to Evaluate Expert System Applications

JILL RAYE HOWES
The Hartford Steam Boiler Inspection and Insurance Company
One State Street
Hartford, Connecticut 06102, USA

ABSTRACT

INTERVIEW is a knowledge based system that evaluates expert system applications. Expert system development often requires significant investments of time and money. Also, there is no guarantee that the project will be successful. Therefore, it is very important to consider the likelihood of success before beginning development.

INTERVIEW helps the user evaluate a project's viability. It considers the management and end user support for the project, the cost benefit, the source of expertise, and the problem type. It identifies potential pitfalls and hazards and suggests solutions. It can also be used to learn more about expert systems and their successful development.

This paper will discuss the considerations involved in potential expert system evaluation and how INTERVIEW can be used to perform this evaluation.

EXPERT SYSTEMS

Today, expert systems are being used in a wide variety of applications. However, many people are still unsure exactly what expert systems are and how they function.

Expert systems are computer programs that emulate human decision-making and judgmental logic. They differ from conventional programs both in the type of problem they address and in the way they solve those problems.

Expert systems are also called knowledge based systems. Although expert is the more popular term, knowledge based is often more accurate. These systems contain expert knowledge, however they are rarely experts in and of themselves. They serve best as assistants to the expert, or as tutors to the non-expert.

Until recently, there was no real shortcut to experience. The investment in time made experientially gained knowledge a key asset for companies of all types. Now, expert systems can be used to capture, preserve, replicate, distribute and standardize knowledge and experience. The potential payoff is an increased return on the investment a company has made in its greatest resource, its people.

Expert systems are often measured by the number of rules they employ to solve the problem. Although definitions of what constitutes a rule differ, these systems can vary in size from hundreds to thousands of rules. Development costs also vary widely and without any apparent correlation to the size of the system. Design News (1) reports that one system of 2,000 rules took approximately 7,000 hours to develop, at a cost of about $1 million. High Technology (2) describes a 100-rule system which took only six weeks and $5,000 to develop. These widely fluctuating costs can be influenced by many factors including: the depth of the knowledge the system addresses, the complexity and consistency of the knowledge, the delivery constraints,

hardware and software costs, and the cost of the original source of expertise. Because they often require such a large investment, expert system applications should be chosen carefully.

The interest in expert systems has resulted, at times, in some unrealistic expectations of their capabilities. Although expert systems can provide significant payoffs in increased profitability or productivity, they can also be very costly to develop. We hear about many successful expert systems; however, we seldom hear about those that fail. Learning more about the realities of expert systems can impact significantly how you approach their development. This knowledge can result in enormous savings when you focus on applications that will provide the greatest benefit.

INTERVIEWING A CANDIDATE APPLICATION

INTERVIEW was designed to help the user consider various expert system applications and choose the ones that have the greatest potential for success. The INTERVIEW program simulates an interview of a "candidate application" to evaluate the feasibility of developing an expert system for that application. At the same time, the user can learn the key issues involved in expert system development as they participate in the interview.

FOUR BASIC CONSIDERATIONS

INTERVIEW checks the suitability of a candidate application in four areas. The four considerations are: Is the proposed problem the right type for optimal expert system development? Will the development effort be cost effective? Is there a suitable source of expertise? Will the project have the necessary support from management and users?

Problem Type And Scope

An INTERVIEW session starts by identifying the problem area or "domain" and the specific task within that domain being considered. This "problem type" portion of the interview determines if the candidate application qualifies by checking three basic criteria: Is the problem domain well-bounded? Is the specific problem well-defined? Does the problem require expertise?

It is important that the application be a well-defined problem within a well-bounded domain.

Understanding the difference between knowledge based systems and conventional computer programs helps emphasize the importance of narrowly scoped applications. Conventional programs know formulas. They "think" or calculate according to a predefined algorithm. They know how to do very specific, repetitive tasks. Knowledge based systems know about tasks. They "think" symbolically. They reason according to what they know about objects and their interrelationships.

As an example, consider starting a car. A conventional computer program KNOWS HOW to start a car by following an algorithm. For instance, check that the car is in park or neutral, put the key in the ignition, and turn the key. If the car does not start, the conventional program ends in failure. An expert system on the other hand KNOWS ABOUT starting cars. It knows about the ignition, the fuel injection system, the electrical system, etc. It knows about their functions and their relationships. If the car does not start, the expert system can investigate the cause of failure and recommend a plan to solve the problem. For instance, it may first check the fuel system, find that the gas tank is empty, and recommend filling the gas tank and then trying again.

Because knowing about things requires much more information than just

knowing how to do things, it is important that the domain be well-bounded. Unbounded problems are likely to fail because of the large amount of knowledge needed to solve the problem. It will be difficult to completely identify and correctly encode all of the necessary information and rules. A large knowledge base has a greater potential for incompleteness and contradictions. Even if the system can be successfully implemented, the time required to process extensive knowledge bases can be prohibitively long.

For a similar reason, the scope of the problem should be well-defined to ensure that all of the problem-solving knowledge can be clearly identified. If the problem is not well-defined, it is unlikely that you will be able to identify the correct problem solution and, therefore, to design a system that will properly address it. Thus the most important first step in building an expert system is clearly identifying the problem and the solution.

Once it has been determined that the problem is well-defined within a well-bounded domain, it is important to determine if it is the right "type" of problem. Some problems are better suited to expert system development, others to conventional programming methods. If the problem is one that is relatively static (unchanging) and can be solved algorithmically, then conventional programming methods can be used for the solution. These methods typically can be implemented at a lower cost and with less effort.

Additional criteria concerning the problem type include: Can partial or imperfect results be tolerated? Does the scope of the problem justify expert system development? Can the problem be broken down into parts which can be individually addressed?

If the problem is one that can have partial or imperfect results, or if it can be solved with incomplete information, it is a good candidate for expert system development. Expert systems can tolerate incomplete data. They are good at determining the best possible

solution, when there is no singularly correct solution.

A problem that can be broken down into parts can be more easily solved and encoded in steps. The knowledge required for each part of the problem solution can be isolated and incrementally built into the system.

Obviously, it does not make sense to build an expert system for a problem too small in scope to justify the effort.

Cost Effectiveness

Before embarking on the development of an expert system, it is advisable to determine the cost effectiveness of the project. INTERVIEW can't perform a true cost benefit analysis; however, it helps stimulate thinking by asking questions about the potential benefits of the system and raises the user's awareness of the costs involved.

INTERVIEW first tries to determine if the need for the task is long-term. This is a natural prerequisite to any effort directed at solving a problem. It is even more important when you realize that expert systems usually cost more and take longer to develop than many alternative solutions. Thus, an expert system may not be the best approach for a temporary or interim solution.

INTERVIEW next looks at the potential payoffs from the system. Such payoffs may stem from direct increases in profitability and productivity, or indirect payoffs through product differentiation. Direct increases in profitability may arise from the sale of either the expert system itself or of the service it provides. Increased productivity is the benefit most common to expert system applications. Expert systems can help novices perform more expertly, and experts more efficiently.

Additional factors which are considered while evaluating the cost effectiveness include: Are any alternative solutions being pursued? How "expensive" is the knowledge to be captured in the system? Will your expert's productivity be increased?

Quite simply, less costly alternative solutions may negate the need for an expert system solution.

The company may benefit greatly from the preservation and distribution of expensive knowledge to less experienced workers. Another benefit comes from freeing an expert from mundane, repetitive thinking tasks to concentrate on more complex problems.

Additional costs include: the cost of the software and the hardware, the cost of the expertise, and the cost of developers time.

Expertise

Expert systems are so-called because they emulate the knowledge of experts. This knowledge must be acquired from an expert, or some other suitable expert source. The knowledge must be analyzed by specially trained knowledge engineers and encoded into the computer system. The knowledge must be of a suitable type for this process.

The third category of questions that INTERVIEW covers concern the source of expertise: Does a source of expertise exist? Is there currently someone capable of solving the problem? If so, is he/she available to spend the time needed to develop the system?

It is not always obvious that the source of expertise for an expert system is crucial. The knowledge engineer must have a source of knowledge to encode in the system. If there is not a suitable source, either the knowledge engineer will need to become an expert in the problem domain, or the system may not correctly solve the problem

It is necessary, not only that there be an expert, or knowledge source, but it is important that the expert be available. The knowledge engineer will need to have access to the expert to glean the knowledge, and then to test the system.

INTERVIEW also tries to determine the "quality" of the source of expertise by asking such questions as: How often is the expert right? How well does the expert communicate? How cooperative is the expert?

The system can't be expected to be any more reliable than the knowledge given to it. There are some obvious factors to consider about the reliability of the knowledge source - such as how often the expert is right, and how often the problem can be solved.

There are also some less obvious factors that contribute to the quality of the knowledge in the expert system. If the expert is unable to communicate effectively, there is a greater potential for error in the rules. Also, poor communication requires a longer development time, more frequent consultations with the expert, and more adjustments to the system.

There are several difficulties with an expert that is unwilling to cooperate, and this potential resistance to the system must be considered. The expert may have a fear of being "replaced" by the expert system. Often, he/she may be uncomfortable with such a systematized examination of their knowledge. Or, the expert's time might be in great demand, and the development of the system is considered an inconvenience.

An alternative source of expertise can be found in books or journals. Although this may appear appropriate initially, in the long run, this method can be less effective. The knowledge engineer may need to become the expert. This introduces the danger that the knowledge in the books be misinterpreted when encoded. Without an expert to examine the system as it is developing - and make note of "exceptions

to the rule" - the system may have little validity.

Once an adequate source of expertise has been identified, INTERVIEW determines whether or not the knowledge is of the right type: Is the knowledge "teachable?" Has the knowledge been acquired through experience or by studying formulas.

If the knowledge cannot be transferred from one person to another, it will be hard to "teach" the system how to solve the problem. If it was acquired by studying formulas, it may be easier to use conventional programming methods to encode those formulas.

Support For The Project

One of the most important considerations in the development of an expert system is whether there is sufficient support for the project.

The hype and the confusion surrounding this new technology can lead to management expectations that cannot be met. These expectations must be controlled, by educating management about the potential problems, and the necessity of their own commitment to the success of the project.

Management can only make this commitment if it is well informed about the costs and complexities involved in developing expert systems. Expert system project time frames are very difficult to quantify, and there is a potential for partial or total failure. Therefore, it is important to have support from management that will endure setbacks in schedule, opposition from adversaries, and even the need to redefine objectives and problem definitions as the problem solving process is examined.

In addition to cost, it is important for management to be well informed of the benefits involved in order to make decisions about the

priority of the project. This is especially true when there are many demands placed upon an expert's time. Management support may be needed to ensure cooperation from these experts and to encourage end users to help in the design stages. They may also need to support the implementation of the system, and help people adjust to changes that it brings to their work process.

INTERVIEW will ask questions about management's knowledge of and support for the project: Is there strong management support for the project? Is management aware of the costs involved? Is management aware that development time frames are difficult to quantify? Is management aware there may be resistance to the project?

Again, without strong management support it is unlikely the project can succeed. Management must be aware of the costs involved and the potential pitfalls as well as the benefits to be able to fully commit to the success of the project.

INTERVIEW also considers the end users support of the project: Have the user's of the system been identified? Have they been consulted? Do they agree there is a need for the system?

It is important that the "audience" be identified and their input considered when designing the system. Otherwise, the end result is unlikely to meet the user's needs. Additionally, it is important that the end users be carefully informed about the realities of the expert system. If they have fears that the system is being designed to replace them, they are unlikely to cooperate or accept the system. No matter how well the system works, it is worthless if it is never used.

HOW THE INTERVIEW PROGRAM WORKS

INTERVIEW consists of two parts: (1) a brief tutorial on expert systems and (2) an expert system that examines an application to

determine if it is a good candidate for an expert system. The INTERVIEW program was designed to be friendly and non-threatening. The program assumes the pseudo personality of ACE -- the Application Candidate Expert, to execute an interactive dialogue between the user and the computer.

The tutorial section covers some basic artificial intelligence definitions, explains what type of candidates make good applications for expert system development and provides more details about the INTERVIEW program itself.

The interview portion of the program evaluates a candidate application and investigates the feasibility of expert system development. During the interview, the user can learn about the considerations involved in making development decisions by asking ACE to explain why questions are being asked.

ACE offers the user a variety of options with most questions. Answers are typically "yes", "no", "maybe", "don't know"; however, a couple are multiple choice options. When asked a question, the user can also respond with the "rephrase" option or the "inform" option. When the user chooses the rephrase option, ACE will rephrase the question. When first using the system this option can be used to both better understand the questions and to learn more about the nuances of the evaluation. When the user chooses the inform option, ACE will explain why a question is being asked. This option is also helpful in learning what qualifications are necessary for good expert system applications. Both options assist the novice user, helping him/her to learn more about expert systems, without slowing down the familiar user while performing a routine interview.

As the interview progresses, ACE will make various observations and recommendations regarding the candidate's likelihood for success. If the application does not meet some important criteria, ACE may end the interview early and ask the user to "reconsider." At the end of the

interview ACE will give a final evaluation of the candidate application. INTERVIEW maintains three files filled with 1) the questions that were asked and why, 2) the warnings ACE has given the user, and 3) the comments ACE has made about the candidate and the final recommendation. At the end of the interview the user can review these files or print them out.

LIST OF SAMPLE INTERVIEW QUESTIONS

What follows is a sample of the questions that "ACE" might ask during an interview. They may not all be asked, and some questions may be asked that are not listed below - it all depends on the answers. If you don't know the answer to a question ACE will probe further with additional questions. When given a definitive "yes" or "no", ACE will usually take your word for it and ask no further questions on the topic.

THE PROBLEM TYPE:

1. Is the problem domain well-bounded?

    1. Could one person learn enough about a field to be an expert across the entire domain?
    2. Does knowledge need to be gathered from many different sources in order to solve the problem?

2. Is the specific problem very well-defined?

    1. Can the important inputs to and outputs from the system be identified?
    2. Can the problem be broken down into subproblems?
    3. Is there a specific solution to the problem?

3. Is the problem one which requires expertise to solve?

    1. Can the solution be found simply by using some formula?
    2. Have conventional (algorithmic) computer program approaches to solving the problem worked?
    3. Does the problem require making decisions?
    4. Does the solution of the problem require the use of various strategies or a choice between various strategies?

4. Can partial solutions or non-optimal results be tolerated?

5. Is the problem large enough to justify the complex process of developing an expert system?

THE COST AND BENEFIT:

1. Is the need for this task expected to be long-term?

    1. How many years is the task expected to continue to be needed?
    2. Are alternative solutions to the problem being pursued?
    3. Is this a new task or has it been necessary for a long time?

2. Is a significant payoff expected from the development of this system?

    1. Is a direct profit expected from the sale of the expert system?
    2. Will there be an indirect profit from the increased ability to sell some product(s) and/or service(s) due to the development of the expert system to accomplish the task?
    3. Is the expert system expected to differentiate some product thus increasing its marketability?

3. Is the cost of the expertise expensive, moderate, or cheap?

    1. Will the expertise be unavailable in the future?
    2. Is the expert's time valuable?
    3. Is the expertise scarce?
    4. Is the expert overworked?
    5. Would freedom from the task allow the expert to devote more of his/her time to more critical tasks?

THE SOURCE OF EXPERTISE:

1. Is there a suitable source of knowledge?

    1. Is there an expert who knows how to solve the problem?
    2. Will the expert be able to commit a substantial amount of time to the development of the system?
    3. Is there more than one expert who will be used as the knowledge source?
    4. Is there another source of knowledge, such as charts, or a written procedure for the problem solving?

2. Is the knowledge source suitable for expert system development?

    1. Is someone with experience better at solving the problem than an amateur?
    2. Can the problem solving knowledge be passed on to an amateur?
    3. Are the expert's solutions reliable? Is his/her judgment trusted?
    4. What percentage of the time does the expert reach a correct solution?
    5. How many years has the expert been solving the problem?
    6. What percentage of the time can the problem be successfully solved?
    7. Is it possible to test the results of the solution?

3. Will the expert be cooperative?

4. Is the expert able to communicate well?

SUPPORT:

1. Is there strong management support for the project?

2. Is management willing to invest the necessary time, money, and energy that will be needed to complete an expert system project?

    1. Is management aware that expert system development time frames are very hard to quantify?
    2. Is management willing to make the necessary monetary investment to build the expert system? (This includes costs of the expert's time, the developer's time, the hardware and the software.)
    3. Is management aware that they may need to expend considerable energy to ensure cooperation with and acceptance of the system?
    4. Does the problem solution lie on the critical path of some other project or process?
    5. Did management identify the need for this task?

3. Has an audience for the system been clearly identified?

4. Have the potential users of the system been consulted? Are they aware of the project and has their input into the solution design been sought?

    1. Does the user agree that there is a need for the expert system?

2. Has the user been consulted to determine how the system can be designed to be most helpful?

OBTAINING INTERVIEW

INTERVIEW is a copyrighted program of The Hartford Steam Boiler Inspection and Insurance Co. It is available to the public upon request.

ACKNOWLEDGMENTS

Much of the knowledge in the INTERVIEW expert system was based on an article by D. Prerau entitled "Selection of an Appropriate Domain for an Expert System." (3)

REFERENCES

1. Gautschi, T.F., "Expert Systems Come Into the Mainstream," Design News, Vol. 43, November 9, 1987, p. 206.

2. Davis, D.B., "Artificial Intelligence Goes to Work," High Technology, Vol. 7, No. 4, April 1987, pp. 16-25.

3. Prerau, D.S., "Selection of an Appropriate Domain for an Expert System," AI Magazine, Vol. 6 No. 2, pp. 26-30.

Development and Application of an Expert System (HITREX) for Plant Operational Support

A. KAJI, T. MARUYAMA, and Y. EKI
Hitachi, Ltd.
5-2-1 Omika-cho, Hitachi-shi
Ibaraki-ken, Japan

ABSTRACT

To construct a useful expert system, a tool to generate and to expand knowledge efficiently is requisite. A fault-tree based expert system shell,HITREX, has been developed to provide easy knowledgebase generation and modification in an on-line environment. It has also a forward and backward inference mechanism and guide message control. The features of HITREX and its application to an operational support system are discussed.

INTRODUCTION

With the increased capacity of plants and a variety of operation schemes, operation of power plants is requiring higher technique. Even in the age of fully automatic plant start-up and shut-down, computer systems can not handle any abnormal conditions of the plant. Therefore,the expertise of well-experienced operators must be generalized and made accessible to any operators.
To solve this problem, we have developed HITREX ,a fault-tree based real-time expert system shell. In the following, the features of HITREX and its application are described.

FAULT-TREE BASED EXPERT SYSTEM SHELL-HITREX

Objective of HITREX Development

To make an expert system truly useful, generation and expansion of knowledge should be easy for any persons. If so-called knowledge engineers or special engineers are required in construction and expansion of knowledgebase, it would be very inconvenient and the knowledge would not be improved.
We have developed HITREX which enables a plant engineer to easily construct an expert system with knowledge expressed in fault-tree form. The design objectives of HITREX development is as follows.

As a knowledge construction tool, HITREX provides;

- o Easy construction of knowledgebase
- o Easy expansion and modification of knowledgebase
- o Easy accumulation of knowledge

As an inference engine, HITREX presents;

- o Dynamical response to actual plant condition
- o High reliability of inference

As a guidance system, HITREX performs;

- o Display of the plant condition in an easily understandable form
- o Dynamic guide message display reflecting the plant condition

To achieve these objectives, we implemented the following features in HITREX.

- o Knowledge construction capability just by drawing fault-trees on a CRT screen
- o Fast modification and expansion of knowledgebase by just compiling modified or expanded part of fault-trees
- o Incorporating analog process values in inference by converting those values into degrees of abnormality
- o Utilization of operator's judgment or information whenever no plant data is available in the computer to estimate abnormality of an event of fault-trees
- o Display of graphical information in either mimic diagram or trend graph form along with guidance messages

Configuration of HITREX

Fig.1 shows the software structure of HITREX. EUREKA-II is a knowledge processing system based on production rules.
It has the following features.

- o Rule and fact type knowledge representation
- o Fast forward reasoning with uncertainty factor
- o Knowledge editor and debugger with explanation function

HITREX converts the knowledge information entered in fault-tree form into rules and frames to be fed into EUREKA-II. Each block of fault-trees is defined by plant parameters using a point identification number which is an index of the plant database adopted in a plant computer. EUREKA-II performs inference referring to plant data through the mapping table automatically generated by HITREX. The result of inference is displayed in suitable form to operators.

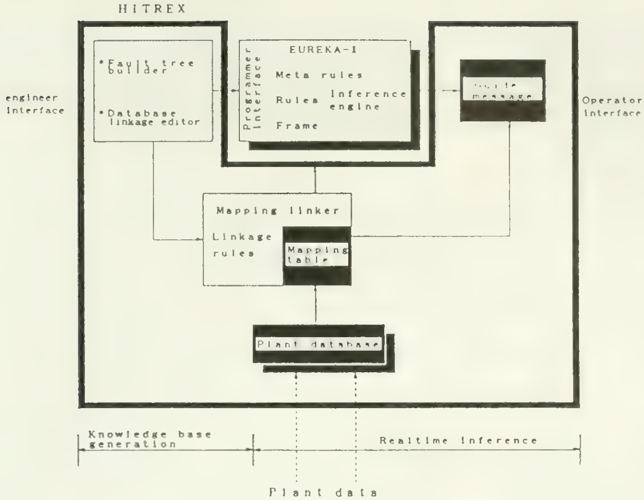


Fig. 1 Software Configuration of HITREX

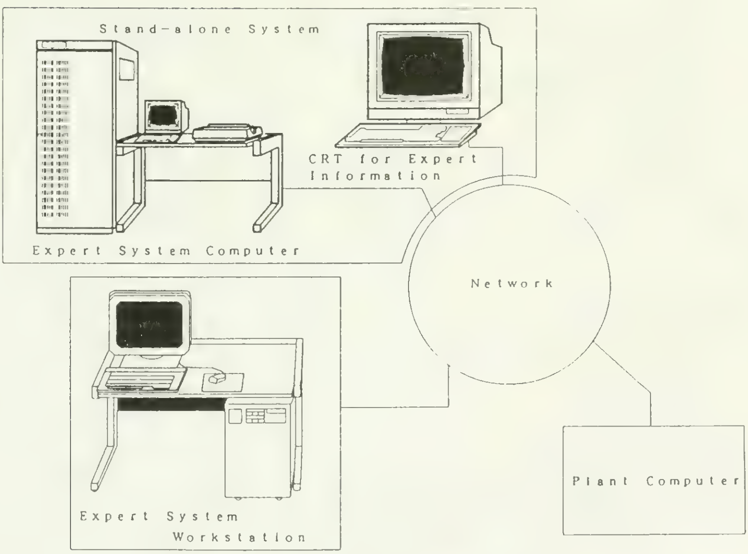


Fig. 2 System Configuration

Fig.2 shows a typical hardware configuration of HITREX. Knowledge construction is done on the workstation. The installed knowledge is compiled and the object module is transferred to the on-line inference computer. The inference computer receives plant data from the plant computer and performs inference based on the knowledge transferred from the workstation. The guide messages are displayed on the CRT which is installed in a control room.

Knowledge Generation

Knowledge generation includes the following steps:

- o Decide the location of the block.
- o Define the abnormality of the block.
- o Define the guide message corresponding to the block.
- o Enter the abnormality propagation factor.

On the screen of the workstation, the new block can be created by just pointing to the location. Fault-trees can be scrolled up and down or left and right using scroll bars. After the definition of location, the abnormality of the block is defined on the formula window using a membership function as shown on Fig.3.1. Then guide messages are defined on the message window (see Fig.3.2). many messages can be defined corresponding to various levels of abnormality. Actual plant data can be embedded in guide messages by assigning the point identification number as "@ point identification number". On the guide message display, actual plant data of that point identification number is displayed where "@ point identification number" is defined. Abnormality of any block propagates to the downstream with confidence factor (abnormality factor) which can be defined on the left-hand window (see Fig.3.3). To easily define the block abnormality, the plant parameter list can be briefed through on the I/O list window shown on Fig 3.4.

On-line Inference Mechanism

HITREX supports forward chaining and backward chaining based on information expressed in fault-trees. The forward chaining is used for plant diagnostic and fault prediction, or more accurately, prediction of fault propagation. The backward chaining is used for alarm analysis and guidance systems. Fig. 4 shows the inference scheme of forward chaining. R_k shows the propagated abnormality from the upstream. Each block, for example block Z_k, calculates its abnormality based on the abnormality definition of the block using a membership function as described on Fig 5. Taking into consideration of the propagated abnormality and its own abnormality of the box, X_k is transferred to the downstream. Therefore, the abnormality obtained by inference is accurate because it is suitably corrected by the actual plant status.
In the backward chaining, abnormality of every path of the fault-trees is calculated to identify the cause of a trouble. If information which is not available in the system is required, such information is asked of an operator and entered information is used for further investigation of trouble cause.

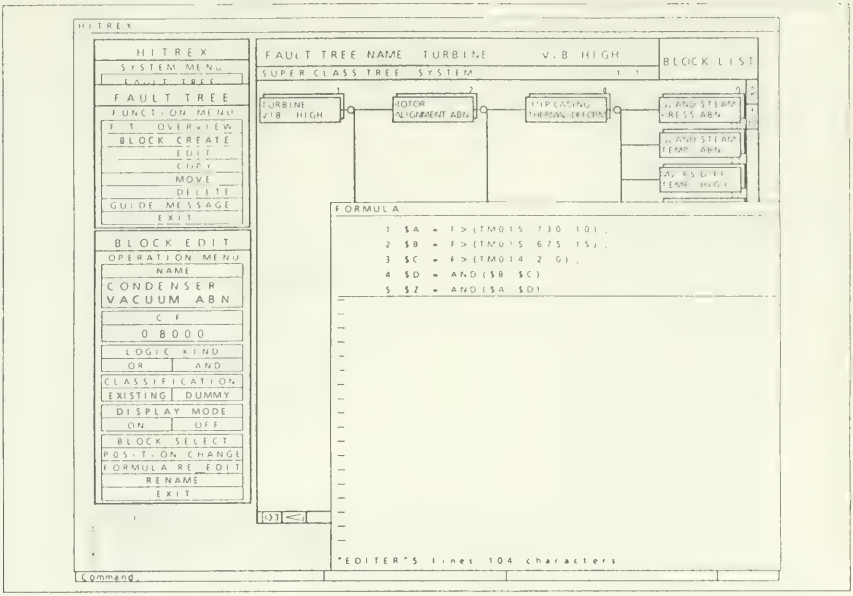


Fig. 3.1 Tree Definition

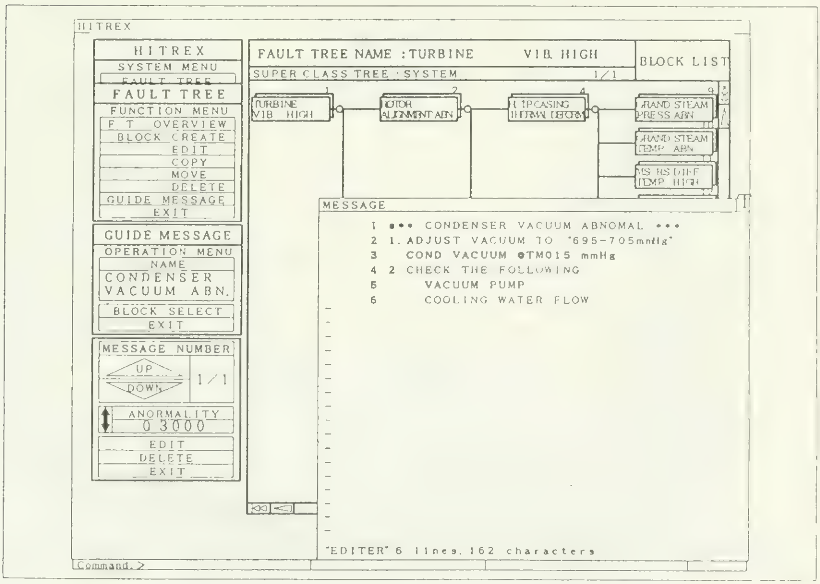
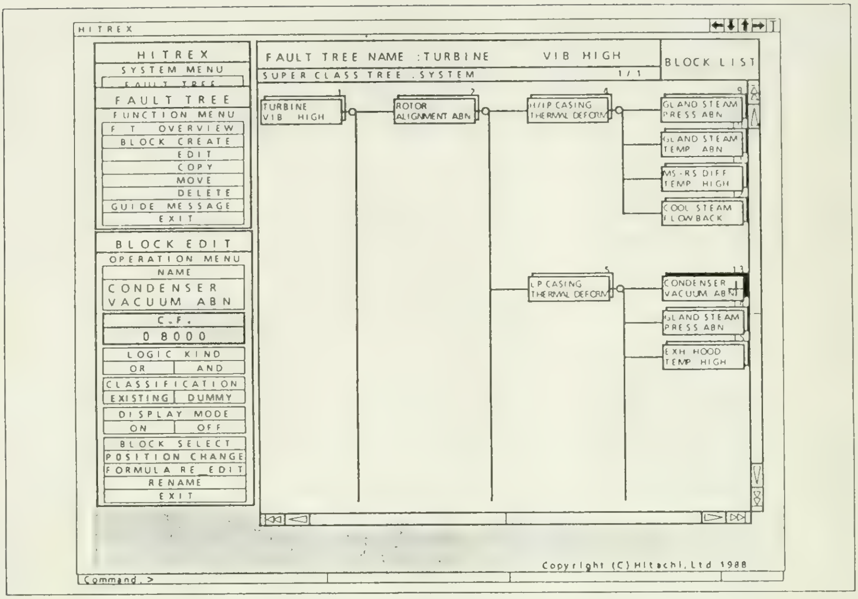
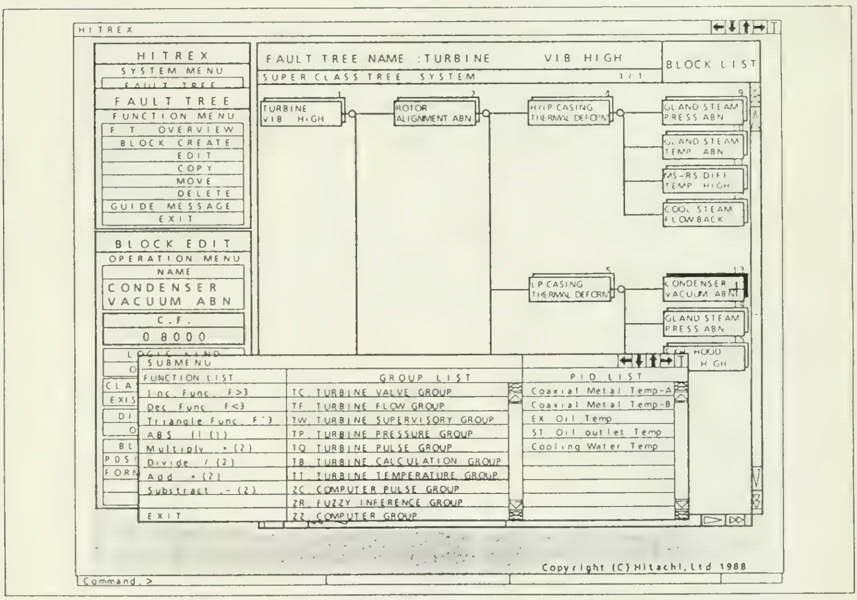


Fig. 3. 2 Guide Message Definition



F i G . 3 . 3 Abnormality Propagation Factor



F i g . 3 . 4 I / O List Window

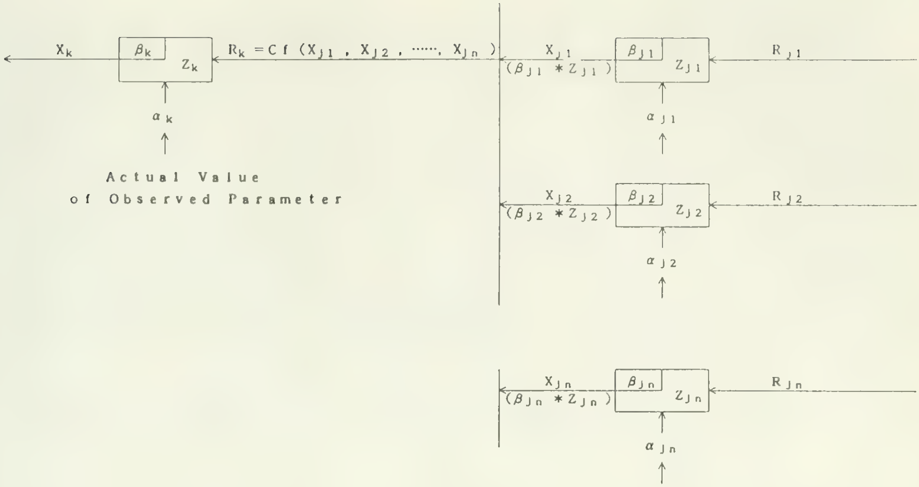


Fig. 4 Inference Algorithm

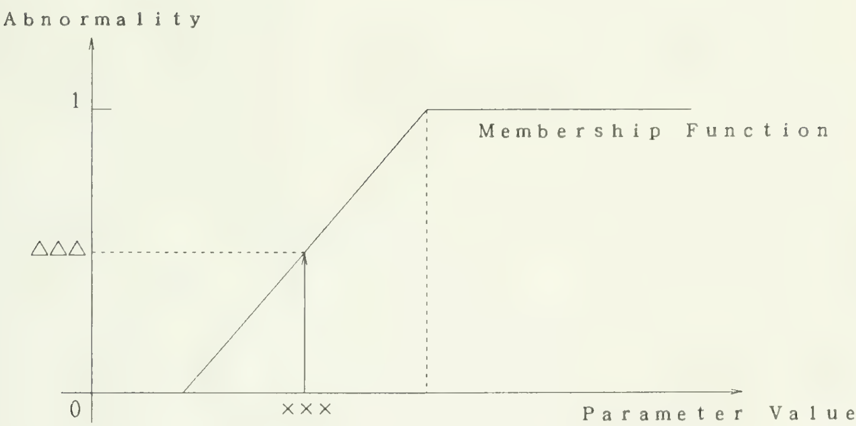


Fig. 5 Membership Function

APPLICATION OF HITREX TO OPERATIONAL SUPPORT SYSTEM

Role of Operational Support System

An operational support system augments monitoring and control functions of the plant computer with a more advanced technique or algorithm than the plant computer. Fig. 6 shows that operational support functions are performed in various systems. The plant computer performs overall monitoring of the plant based on limit checking or correlational consistency check of plant parameters. On the other hand, the equipment diagnostic system is equipped with special signal processing like FFT (Fast Fourie Transform) or AE (Acoustic Emission) processing and execute diagnostics dedicated to each equipment. The turbine vibration monitoring system is an example of an equipment diagnostic system. An operational support system takes care of unit level processing based on advanced algorithms such as inference on knowledgebase, and fuzzy inference. Typical functions of this system are as follows.

- o Alarm analysis and guidance based on AI
- o Plant start-up scheduling optimization based on fuzzy inference
- o Optimization of plant control
- o Plant overall monitoring based on AI

The relation between an operational support system and an equipment diagnostic system is analogous to the relation between a plant computer and a control system.

Alarm Analysis and Guidance System

Whenever an alarm occurs and its corresponding annunciator window is flashed, operators are required to take suitable actions to avoid worsening of the trouble and are also required to repair the cause. This is not easy for any operators. Therefore, an operational support system utilizing the expertise of experienced operators becomes necessary. We are planning to implement an alarm analysis and guidance system based on HITREX. The system contains information on alarm analysis in fault-tree form. Fault-trees corresponding to more than one hundred annunciator alarms will be implemented in the system. When a block for an annunciator becomes abnormal, inference to identify the cause of the alarm starts. Fig. 7 shows the alarm analysis flow based on fault-trees. To find the route to the cause of trouble, the abnormality degree of any route is calculated and compared. If there is no information required from an operator, the system displays guide messages as shown on Fig. 8. The action window shows an action to be taken against the alarm. Guide messages defined for this alarm block appear in the action window. The cause is displayed with the confidence factor. The cause window displays guide messages to check and to repair the cause of the trouble. These messages are those defined for the block corresponding to the cause. On the part of the screen trend curves associated with the alarm are displayed. The trend curves can be replaced by an associated mimic diagram by poking the software-key on the bottom of the screen.

SYSTEM	Level of Support	Operational Support Technique
Operational Support System	Unit level	Complicated processings (AI processing, Fuzzy calculation)
Plant Computer	Unit level	Information processing (Validity check, performace calculation)
Equipment Diagnostic System	Controll-er Level	Signal processing (Vibration analysis, AE processing)

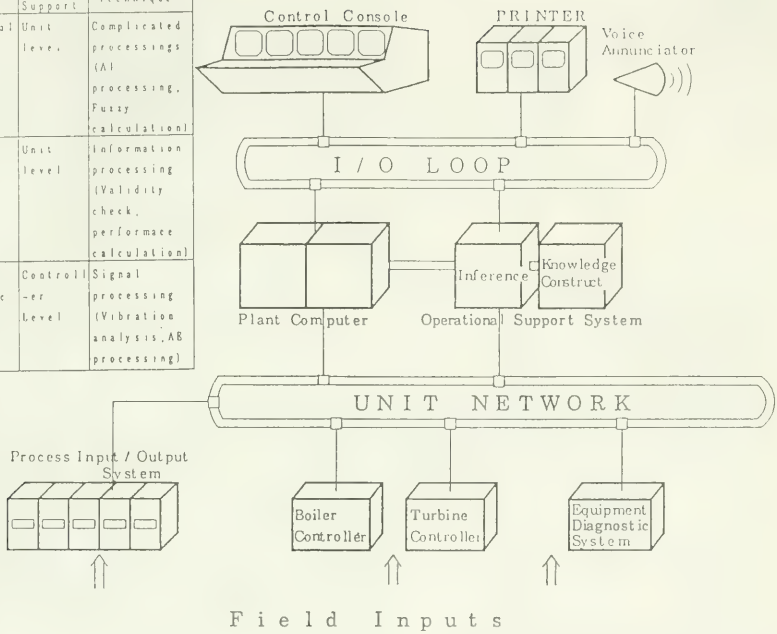
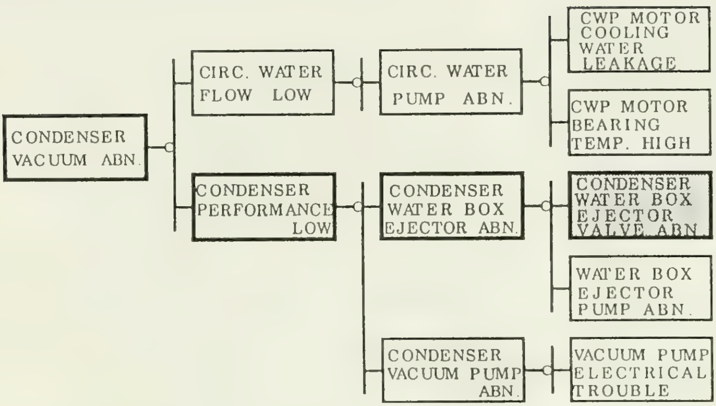


Fig. 6 ROLE OF OPERATIONAL SUPPORT SYSTEM

EXAMPLE OF FAULT TREE



ALARM ANALYSIS FLOW CHART

ITEM REQUIRING OPERATOR'S INFORMATION
FAT BOXES SHOW PATH TO THE CAUSE

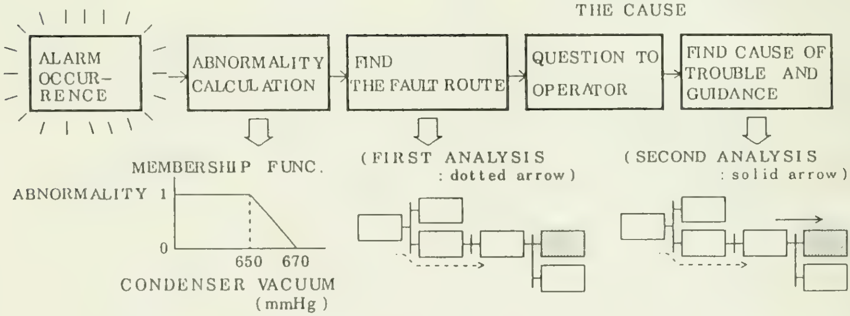


Fig.7 Backward Inference for Alarm Analysis

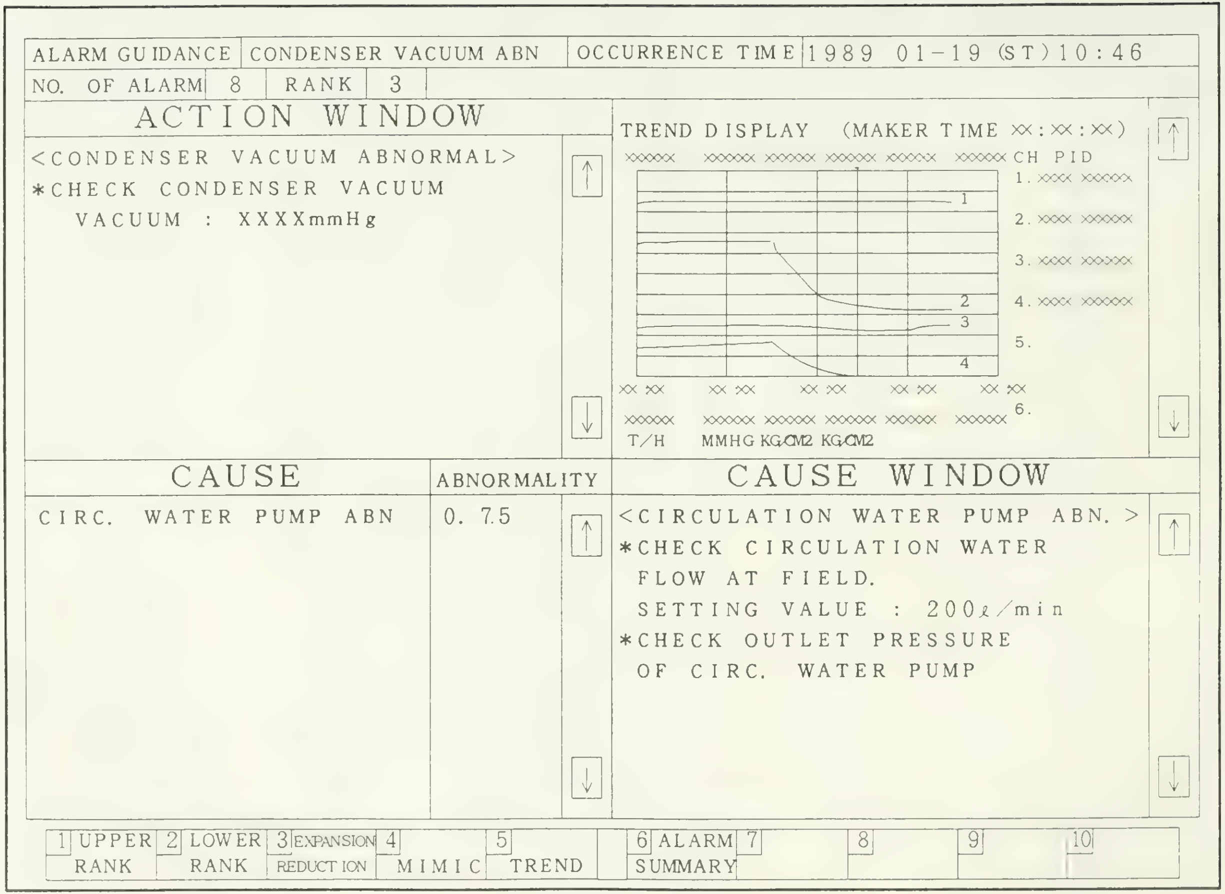


Fig. 8 Example of Guide Message

If information from an operator is necessary for the system to investigate further the cause of the trouble, the system displays a question to an operator. After receiving all answers from an operator, the system again starts chaining based on new information. The ultimate cause of the trouble will then be shown on the CRT.

CONCLUSIONS

HITREX, a fault-tree based expert system shell has been developed to realize an operational support system. HITREX has the following features:

- o Knowledge generation based on fault-trees
- o Incorporating process parameters into inference
- o Guide messages with graphical information on plant status

The most important factor of an expert system is knowledge itself. Even if an expert system shell is good, the expert system will be useless without good knowledge implemented. We hope that HITREX will make a truly useful expert system with implementation of expertise.

REFERENCE

1. K. Aoyagi et al., "A Scheduling Support System Based on Fuzzy Inference for Startup of Fossil Power Plants", Seminar notebook: Expert Systems Applications in Power Plants, EPRI, May 1987

2. A. Kaji et al., "Advanced Features of a Computer System for Thermal Power Plants", Hitachi Review 37-4, Aug. 1978

TURBOMAC: Networked Delivery of Problem-Solving Knowledge

BENJAMIN KLIMCZAK
The Hartford Steam Boiler Inspection and Insurance Co.
Research and Develoment
One State Street
Hartford, Connecticut 06102, USA

ABSTRACT

This paper presents TURBOMAC™ and The Hartford Steam Boiler Knowledge Network Computer™. TURBOMAC is an on-line diagnostic expert system developed and delivered by The Hartford Steam Boiler Inspection and Insurance Company. It provides immediate help for vibration problems in large rotating machinery.

Hartford Steam Boiler developed TURBOMAC over a three-year period as a part of its wide-ranging efforts in loss control. The paper describes this development motivation and TURBOMAC's current user community. These users include more than fifty utility, petrochemical, and municipal insureds throughout the United States.

The paper discusses TURBOMAC's internal knowledge base of vibration symptoms, diagnostic rules, and diagnoses. The paper presents a typical TURBOMAC session and explains how the expert system uses its knowledge to support its conclusions.

TURBOMAC is the second expert system in a library of knowledge based tools available to insureds of the Hartford Steam Boiler on the company's Knowledge Network Computer. The paper briefly describes the telecommunications software Hartford Steam Boiler provides to connect the customer's local PC to the Hartford, CT-based Knowledge Network from anywhere in the world. Using this software, users can access this knowledge library twenty-four hours a day, seven days a week.

INTRODUCTION

Throughout its 123 year history, The Hartford Steam Boiler Inspection and Insurance Company has realized the importance of applying advanced engineering to help businesses and institutions with the reliable and efficient use of property and equipment. This goal has led the Company to employ the latest knowledge-based computer technology. New developments in the science of artificial intelligence allows computers to process ideas instead of just numbers.

The focus of this effort has been the Hartford Steam Boiler Knowledge Network Computer. This computer system is an integrated collection of computer hardware, software, telecommunication and network devices. The system is programmed by a team of software and knowledge engineers to bring problem solving models to the real world of Hartford Steam Boiler insureds. An example of this is TURBOMAC, an expert system for diagnosis of large rotating machinery.

Knowledge Network programs like TURBOMAC serve as high-tech consultants, helping non-experts diagnose potential accident producing conditions. They are based on the expertise of specialists and nearly duplicate the experts' logic and thought processes. These programs serve as silicon-based addition to Hartford Steam Boiler's decades-old system of providing machinery solutions to its insureds. Just like its counterparts in the company's inspection and engineering force, this network is a means of delivering useful experience to customers throughout the world. Instead of using on-site visits, the Knowledge Network layers hardware, operating systems, databases, development environments, communications software, and remote output devices to advise users with machinery and property problems. This design is represented by a block diagram (Figure 1).

To achieve this purpose, the Knowledge Network Computer provides two distinct environments. First, the network includes several developmental workstation environments for acquiring, coding, and testing knowledge for new expert systems. Second, the larger computers of the Knowledge Network allow concurrent delivery and maintenance of existing expert systems to insureds at multiple remote locations.

Network Hardware and Operating System

The Knowledge Network Computer system originates behind glass doors in an air-conditioned room located in Hartford, CT. Here several computers continually run information-based applications and expert systems like TURBOMAC (Figure 2). These computers comprise the heart of the Knowledge Network. The largest has more than 100 megabytes of internal memory and gigabytes of disk storage.

The various storage and processing devices on the Knowledge Network Computer are interconnected via an Ethernet network. The computer network also supports numerous internal terminals and several networked microcomputers, including IBM compatibles and Apple Macintoshes..

The computers and associated networking equipment provide automatic redundant service to remote users. If the primary processor is not available, its programming load and disk storage switches to one of the other computers. The remote TURBOMAC user calling the system would never be aware of the difference. During a recent 14-month measuring period, the Knowledge Network Computer was operational 99.6% of its scheduled run time.

The communications servers (Figure 2) are the network devices that allow the processors to be transparently redundant. These devices link the modems accessed by remote users through the Ethernet network to the system's processors. The servers are rack mounted modems in large air-cooled communications cabinets. The various modems operate at data speeds ranging from 300 to 19,200 bits per second. The communication servers can allow up to 72 concurrent remote users to access the Knowledge Network through a single toll-free number. A protocol analyzer allows quick trouble shooting of the communications system. The Knowledge Network's voice synthesizers also reside in these cabinets.

Hartford Steam Boiler's Research and Development staff is constantly evaluating the Knowledge Network's performance and adjusting computing resources to meet demands. For instance, a special network monitor provides periodic reports on each of three types of remote access: those sessions originating internationally, those from within the USA, and those from within Connecticut. The monitor also indicates which communication server port the remote user accessed. This information helps system management staff to quickly adjust system resources to meet peak demands.

KNOWLEDGE NETWORK COMPUTER ARCHITECTURE

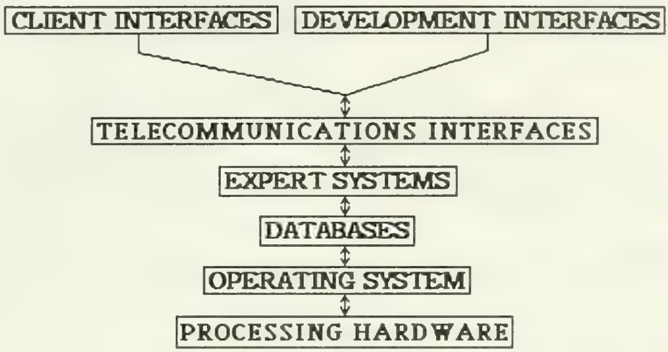


Figure 1. The Knowledge Network Computer is a layered architecture of software and hardware. The layers are transparent to the client who accesses the network through their remote equipment. The client selects from a library of expert systems, which in turn call upon the databases and operating systems of the network. the network provides both client and developmental user interfaces. It also allows a variety of access modes.

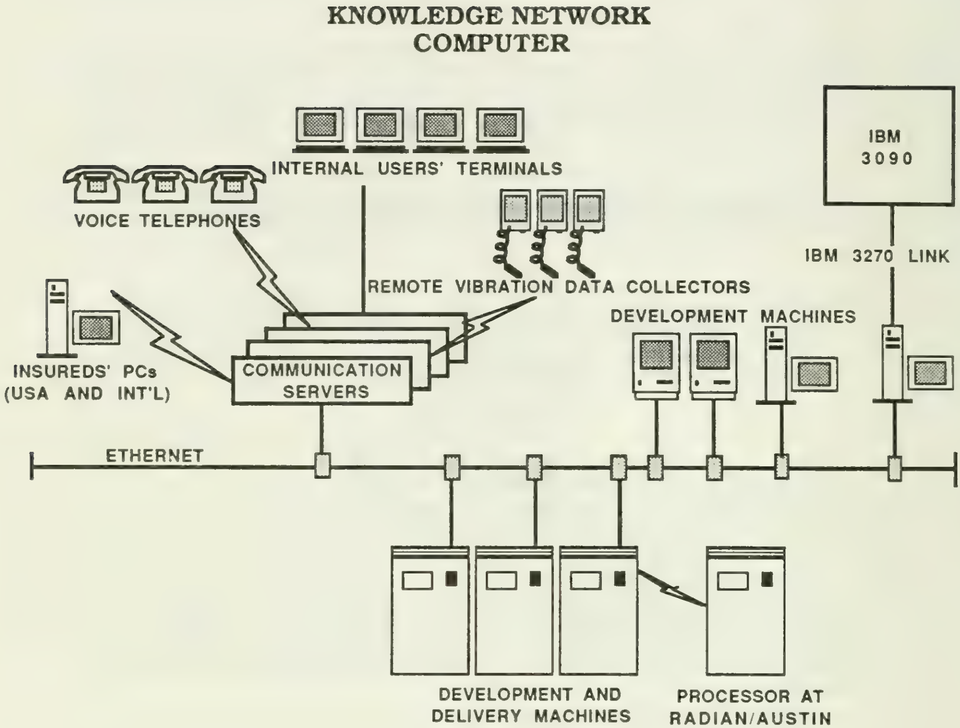


Figure 2. The Knowledge Network Computer Schematic. The communication servers allow different devices to access the network. The network allows expert systems to migrate easily from development to delivery.

User Interfaces

The Knowledge Network provides three kinds of interfaces for remote users. The first involves insureds running expert systems on the Knowledge Network Computer using their local computers and modems. The second kind uses the Knowledge Network voice synthesizing capabilities. Lastly, remote vibration sensors can also automatically access the system's databases.

Insureds accessing the Network's on-line expert systems need only a personal computer, modem, and telephone line. Hartford Steam Boiler provides MS-DOS based or other type of software to its insureds that allows a wide variety of personal computers to automatically dial the Knowledge Network from anywhere in the world. The software asks the user only questions such as whether they have touch tone or pulse service and what numbers are required to obtain an outside line.

The software automatically finds the PC's modem, configures it with the correct baud rate, parity, and stop bit settings, and gives it two numbers to dial the Knowledge Network. If the first number fails, the program directs the insured's PC to try the second one.

When the insureds finish the dial-up and log in procedure, they see a menu screen (Figure 3). This is a top-most view onto the layered resources of the Knowledge Network. This menu displays items of two types. The first group are expert systems and other information or knowledge-based applications. The second group supports system utilities such as network mail, document transfer, or password modification.

Hartford Steam Boiler clients run the communications program using their local computers and modems to gain access to TURBOMAC and the library of other Knowledge Network expert systems . A second type of Knowledge Network program allows users to call with a touch tone voice telephone. The Knowledge Network voice synthesizers can greet the caller and prompt for specific numerical input. The computer can then process these queries against company databases of engineering, account, or claims databases. Voice synthesis has proven successful in providing agents with loss profiles based on occupancy type and zip code from the company's claims records. A voice-based program for property loss control is also being considered.

The third type of access to the Knowledge Network is through remote vibration sensing devices (8). The client brings these small data gathering devices to many points on critical machine trains and gathers vibration readings with a hand-held probe. The devices are programmed to automatically dial the the Hartford computer through conventional phone lines. The Knowledge Network Computer answers these calls and loads the vibration data from the vibration collectors into a database. These readings are available for later plotting, analysis by an experienced expert, or expert system processing.

Knowledge Network Databases

The Knowledge Network expert systems and other applications are built upon company databases (Figure 1). These databases use a relational database technology that allows data to be manipulated in a way closely parallel to its use in the company's information stream.

Over the past several years, relational database management systems like those on the Knowledge Network Computer have become a widely accepted way to manage data. Relational systems offer benefits over former hierarchical and network models in such areas as: (3)

- easy access to all data

- flexibility in data modeling
- reduced data storage and redundancy
- independence of physical storage and logical data design
- a high-level industry standard data manipulation language (SQL)

The database software layer also allows concurrent transactions to occur to the database with no loss of data integrity. It also gives expert systems information quickly assembled from multiple data tables.

TURBOMAC

Expert systems are playing an increasingly important role in the day-to-day solution of real-world problems. (1) TURBOMAC has been available to Hartford Steam Boiler clients for more than than a year. In a typical month, electric utilities account for 42% of the processor time used by TURBOMAC.

What Is TURBOMAC

Diagnosing major faults in large high-speed machines is a complex problem. Because these machines often are responsible for many dollars in daily production, their failure is of serious concern to company management. Since abnormal vibration is a symptom of many types of failure, it is the focus of attention during diagnosis. When abnormal vibration occurs, knowledgeable people immediately must investigate the cause and make informed decisions concerning the repair and future operation of the machine. The timeliness of such diagnosis is critical, as a delayed or an incorrect decision could seriously harm the valuable machine and in turn impact company profitability.

Though sophisticated instruments exist for measuring vibration of rotating equipment, there is a shortage of diagnostic experts able to accurately and rapidly relate the output of these instruments to the cause of the problem.(2) Many facilities have no experts on-site and few, if any, have experts in sufficient numbers to be on duty continuously.

Because of this need and its interest in machinery problem-solving, Hartford Steam Boiler decided to develop TURBOMAC in 1984. After three years of development and testing by software and mechanical engineers, TURBOMAC is now a delivered software product.

Since early 1988, the expert system has given Hartford Steam Boiler clients the knowledge and "rules of thumb" gained from condensing years of machinery diagnosis experience. While the depth of knowledge and experience accumulated by an engineer during a long professional career can never be totally captured, TURBOMAC gives a significant fraction of expertise in an inexpensive and widely available form.

As of March of 1989, 91 sites of 62 companies are using TURBOMAC. These include 55 electric utilities, 12 refining or chemical operations, and 12 municipal power systems. An average session lasts for approximately 67 minutes. Judging from March of 1989, electric utilities account for 42% of the processor time used by TURBOMAC. Several users have provided feedback concerning the benefits they derived from using the expert system (Appendix 3.)

Expert systems, such as TURBOMAC, function both as a screening tool and as experts' assistants. It often permits accurate analysis of a problem without calling in an expert. In these cases, the expert is free to deal with more difficult and costly problems.

Most importantly, expert systems like TURBOMAC provide a way to preserve and use knowledge and experience. The rules of thumb which took vibration analysts and rotating equipment engineers years to develop are available to the entry-level users. Novice engineers can now access the hard-learned knowledge and experience which had been undocumented.

Building TURBOMAC

The real work of TURBOMAC development began only after the problem of machinery diagnostics was identified as an expert system with sufficient benefit to Hartford Steam Boiler and its clients. Software engineers at Radian, Hartford Steam Boiler's engineering and consulting subsidiary, built the knowledge base of the system using RuleMaster (4), an expert systems development toolkit. RuleMaster provided a software framework for constructing and operating a sophisticated knowledge-base.

TURBOMAC was developed with the following six steps (5):

1. The knowledge engineer exactly defined the purpose of TURBOMAC. The knowledge engineer also determined the domain of human expertise to be incorporated into the system.

2. The knowledge engineer documented the expert's information requirements, specialized knowledge, and decision making procedures. The knowledge engineer designed the structure of the expert's knowledge for the system.

This knowledge acquisition step is frequently a difficult stage in the development of any expert system. The domain expert may have difficulty in organizing and verbalizing important knowledge. The knowledge engineer must guide the expert in this effort. To simplify this step during TURBOMAC development, John Sohre's diagnostic charts (6) were used as an initial pattern and source of expertise.

3. The knowledge engineer entered the expert's knowledge, information requirements, and procedures into the computer to create the expert system.

The information provided by the expert is seldom in a form suitable for direct entry into the expert system. In TURBOMAC development, RuleMaster facilitated this work by inducing diagnostic rules from examples supplied by the expert. An expert system rule is a decision point that considers one or two small pieces of data and produces a specific output. A rule is said to "fire" as either conditions in the problem itself or conditions created by previous rules are positively met . The rule then performs its specified action, which in turn "fires" other rules and moves the expert system closer to its conclusion.

4. The expert and the knowledge engineer tested the expert system until it performed satisfactorily. This is done by comparing its performance with the performance of experts using real world data.

5. The knowledge engineer and the expert improved the expert system in those cases where it performed poorly. The validation step repeated until the TURBOMAC reached the desired level of expertise.

6. The knowledge engineer tested the system by monitoring users' ability to provide data to interpret and use the output. TURBOMAC became a practical expert system in this step.

TURBOMAC was originally implemented on a SUN 3/160. Since February of 1988, it has been available to Hartford Steam Boiler clients as part of the Knowledge Network Computer expert system library. The latest update (Version 4.22) was released in December of 1988 and contains 112,197 lines of source code, 1.4 megabytes of compiled code, and 9,700 rules.

How TURBOMAC Works

TURBOMAC diagnoses problems in three major steps (Figure 4). TURBOMAC users begin their consultation by accessing the Knowledge Network Computer with their local PC and modem. TURBOMAC's user interface can be used over phone lines at 1200 baud as it requires only single keystroke operation and eliminates any redundant and time-consuming screen refreshes.

TURBOMAC first allows the user to optionally read on-line instructions to run the program and an explanation of how the system works. With screens like Figure 5, TURBOMAC accepts any subset of 139 problem symptoms For each symptom, the user places an "x" or an asterisk "*" in a column marked "yes", "no", "maybe" or "don't know" TURBOMAC interprets these responses as follows:

yes	the symptom is definitely present
no	the symptom is definitely not present
maybe	the symptom is questionable
don't know	the symptom is unknown

Each line on the question menus (Figure 5) has a corresponding explanation screen (Figure 6). If a TURBOMAC user needs help with how to answer a particular question, an appropriate explanation screen comes up with a "?" keystroke. The system assumes a "don't know" answer when there is no input for a particular symptom.

TURBOMAC Rules

TURBOMAC rules work internally with the values of yes, no, maybe or don't know for each of the symptoms. Based on these answers and the system's rules, TURBOMAC creates two separate scores as it determines its final diagnosis for its consultations: a positive score for each diagnosis that supports its existence, and a negative probability to indicate its absence.

As TURBOMAC applies its rules, it adjusts each diagnoses' positive and negative scores accordingly for each of the user's answers. TURBOMAC especially "looks" for patterns of present and absent symptoms within the user's data. It performs this rigorous matching phase with much greater accuracy than its human counterpart.

A "don't know" answer to a particular symptom tells TURBOMAC the least about the problem at hand. This answer adds no weight either to the positive or to the negative scores. A "yes" answer to multiple symptoms is particularly informative because of the pattern matching feature of TURBOMAC's processing. Since a particular symptom can be caused by many problems, it is not possible to associate the presence of just one symptom to any one of its possible causes.

As TURBOMAC applies its rules to the user's answers, it associates the absence of a particular symptom in a negative way to the appropriate diagnosis. Differences in information content associated with the presence or absence of a symptom is the basis for using two separate scores. Both scores are necessary as TURBOMAC may not be given sufficient information to utilize only negative scores in the diagnosis.

Because TURBOMAC takes symptoms and reasons toward diagnoses, it is considered "forward chaining." It does allow its users to apply "backward chaining" (going from diagnoses back to symptoms) with an option known as "INDICATORS". With this option, the user can learn the major symptom patterns TURBOMAC uses to identify diagnoses.

Because of the general nature of an expert system which must consider more than 139 symptoms and more than 40 machinery faults, TURBOMAC's rule processing accounts for the largest part of the system's coding efforts. However, because of the Knowledge Network Computer's capabilities, remote users typically execute the rule processing within few seconds.

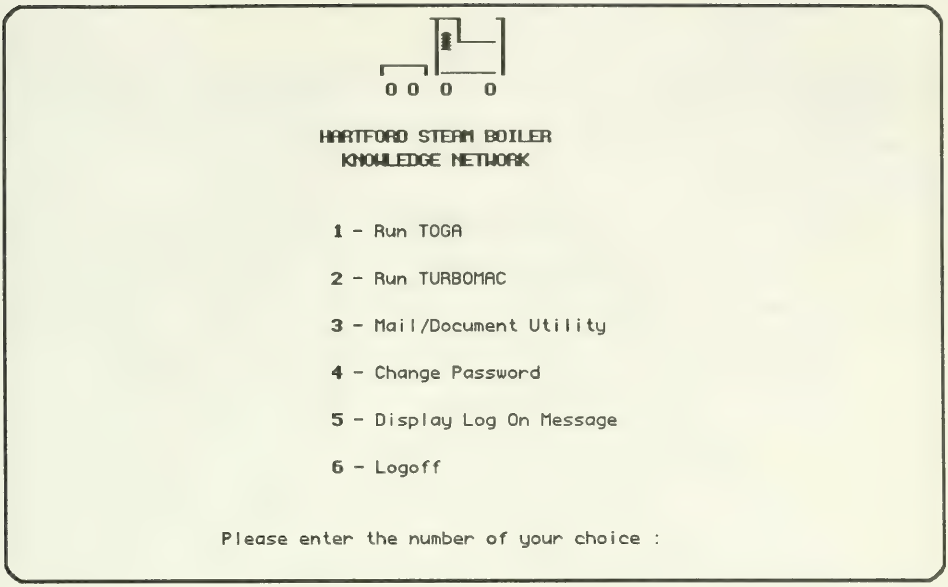


Figure 3. The Main Menu

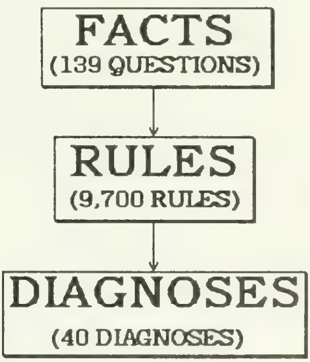


Figure 4. TURBOMAC Processing. TURBOMAC processes the user input with thousands of rules to produce diagnoses and their explanations.

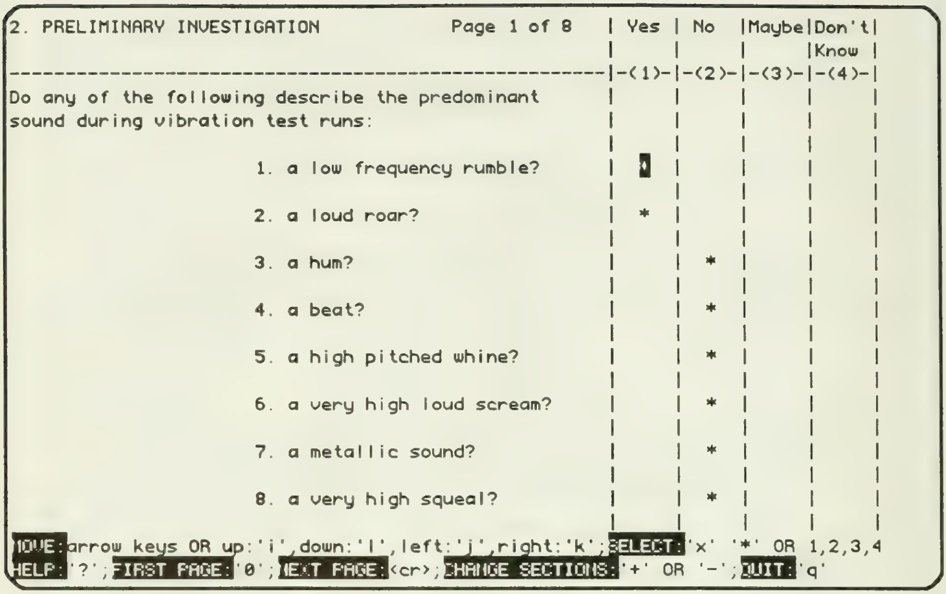


Figure 5. TURBOMAC Question Screen

```
2. PRELIMINARY INVESTIGATION               Page 1 of 8    | Yes | No  |Maybe|Don't|
                                                          |     |     |     |Know |
-------------------------------------------------------------------------------------
| Low Frequency Rumble Sound                                                        |
|                                                                                   |
| Answer YES if the machine in question is emitting an uncharacteristic             |
| low frequency sound.  The low frequency rumble can be difficult to hear.          |
| Often a low frequency rumble will be felt through feet and body.                  |
| The noise has been called a "growl" by operators of 4 pole utility                |
| turbines.  The sound will often have a beating characteristic over                |
| a short period of time, but continues over longer periods.  The sound             |
| often gives the impression of rapping or churning.  Noise is often, but           |
| not always associated with load or speed increases.                               |
| DON'T FORGET EAR PROTECTION!                                                      |
|                                                                                   |
| Answer NO if the machine emits no unusual low frequency sounds.                   |
|                                                                                   |
| Answer MAYBE if there is a low frequency rumble present,                          |
| but is not particularly strong.                                                   |
|                                                                                   |
| Answer DON'T KNOW if you did not observe the sound, or you                        |
| cannot get close enough to hear, or cannot tell the source of the sound.          |
|                                                                                   |
-----------------------------------------------------Hit any key to continue---
```

Figure 6. TURBOMAC Explanation Screen

TURBOMAC Diagnosis

TURBOMAC displays the information it obtained during its rule processing as the final step in a consultation. It develops a diagnostic statement for each of the diagnoses the system knows about. The user can request that TURBOMAC present the reasoning it used for each diagnosis.

Nine possible diagnostic statements are possible, ranging from statements like "Unbalance is very likely" to "Unbalance is very unlikely." TURBOMAC also indicates when insufficient information is available to make any diagnosis.

At the user's request, TURBOMAC can investigate its reasoning for reaching a particular conclusion. This allows the user to recheck any symptoms that appear as important contributions to a conclusion.

Example

The following case demonstrates TURBOMAC' being tested against expert opinion. A 670 megawatt turbine generator experienced some difficulties. The information was entered into TURBOMAC. The diagnostic results of the TURBOMAC analysis are included in Appendix 1.

From those results it is possible to see that a strong indication exists for problems related to critical speed or resonance. One of the benefits of an expert system is its ability to explain its line of reasoning. Explanations of two of the most likely problems are included in Appendix 2.

These explanations begin with COMMENTS which give a general background for the problem. Then SYMPTOMS SUPPORTING THIS DIAGNOSIS are presented followed by SYMPTOMS COUNTER TO THIS DIAGNOSIS. These symptoms and the other information provided in the fault explanation are valuable when courses for corrective action are being determined.

This case was chosen for the amount of information available at the completion of its analysis. It might not be typical of the average problem analyzed by the system in the field. It was felt, however, that it would exercise the system to its fullest potential.

TURBOMAC agreed well with the analyst in this example case. A thorough comparison of the analyst's and TURBOMAC's results in this and other cases has been separately published. (7)

CONCLUSIONS

Responding to a growing time-critical need for solutions to machinery and property problems, Hartford Steam Boiler has created the Knowledge Network Computer. TURBOMAC, an expert system for the diagnosis of vibration problems in large rotating equipment, is part of a library of knowledge based programs available on the Network.

Through use of current technology like the Knowledge Network Computer, expert systems can help solve complex problems. End users no longer need invest in highly sophisticated and expensive hardware and software platforms. Users also need not be sophisticated computer users or telecommunications experts to get the on-line twenty-four hour support of expert systems designed for real-world problem solving.

ACKNOWLEDGEMENTS

The following personnel of Hartford Steam Boiler made valuable contributions to this paper: Napolean W. Benaitis, Computer Coordinator; Kenneth D. Bever, Knowledge Engineer; Peter S. Moon, Manger of the Technical Resource Center; Stephen T. Szymanski, Manager of Expert Systems Development; Donald A. Smith, Director of Research and Development.

REFERENCES

1. Moore, T. "Pushing Performance With On-Line Monitoring." *EPRI Journal*, Vol. 13, No. 6, Sept. 1988, pp. 5.

2. U. S. Department of Labor, Bureau of Labor Statistics, Washington, DC. *Occupational Outlook Handbook, 1988-1989 Edition.* pp. 56-57.

3. Codd, E.F. "Relational Database: A Practical Foundation for Productivity." *Communications of the ACM*, Vol.25, No.2, Feb.1982, pp.109-117.

4. Michie, D., Muggleton, S., Riese, C., and Zubrick, S. "RuleMaster: A Second Generation Knowledge Engineering Facility." *Proceedings of the First Conference on Artificial Intelligence Applications*, Denver, CO, December 5-7, 1984, pp. 591-597.

5. Parge', P., Stuart, J., Vinson, J. "The Development and Application of TURBOMAC, an Expert Machinery Diagnostic System." Presented at the EPRI Seminar on Expert Systems Applications in Power Plants, Boston, MA, May 27-29, 1987.

6. Sohre, J. S. "Turbomachinery Problems and Their Correction." *Sawyer's Turbomachinery Maintenance Handbook, Volume II*, Norwalk, CT: Turbomachinery International Publications, 1980, Chapter 7.

7. Parge', P., Stuart, J., Vinson, J. "Application of TURBOMAC, An Expert Machinery Diagnostic System." *Proceedings of the Vibration Institute Tenth Annual Meeting*, Las Vegas, NV, June 24-26, 1986, pp. 139-146.

8. Bever, Kenneth D., Raymond, Thomas M. "DATALERT: A New Generation of Automated Predictive Maintenance Systems" To appear in 1989 ASME Industrial Power Conference, October 1989.

APPENDIX 1. PROBLEMS DIAGNOSED BY TURBOMAC ORDERED BY LIKELIHOOD

PRESENT
Critical Speed (P4)

VERY LIKELY
Temporary Rotor Bow (U3)
Bearing Damage (B2)
Rotor and Bearing System Critical (C2)
Coupling Critical (C3)
Overhang Critical (C4)
Rotor Rub - Axial (D4)
Resonant Vibration (P5)

LIKELY
Structural Resonance of Casing (S1)
Structural Resonance of Supports (S2)

POSSIBLE
Piping Forces (D6)
Thrust Bearing Damage (B5)
Aerodynamic Excitation (C1)

EQUIVOCAL
Coupling Inaccuracy or Damage (G2)

EQUIVOCAL (A)
Unbalance (U1)
Permanent Bow or Lost Rotor Parts (U2)
Seal Rub (D3)
Journal and Bearing Eccentricity (B1)
Insufficient Tightness in Assembly of Rotor (T1)
Insufficient Tightness in Assembly of Bearing Liner (T2)
Insufficient Tightness in Assembly of Bearing Case (T3)
Structural Resonance of Foundation (S3)
Pressure Pulsations (M1)
Friction Induced Whirl (P3)
Clearance Induced Vibrations (P9)
Torsional Resonance (P10)

UNLIKELY
Bearing and Support Excited Vibration (B3)
Unequal Bearing Stiffness - Horizontal vs Vertical (B4)
Insufficient Tightness in Assembly of Casing and Support (T4)
Gear Inaccuracy or Damage (G1)
Vibration Transmission (M3)
Oil Whirl (P6)

VERY UNLIKELY
Dry Whirl (P8)
Transient Torsional (P11)

ABSENT
Casing Distortion (D1)
Foundation Distortion (D2)
Misalignment (D5)
Sub-Harmonic Resonance (P1)
Harmonic Resonance (P2)

Resonant Whirl (P7)

NOT APPLICABLE
Electrically Excited Vibrations (M2)
Oil-Seal-Induced Vibration (M4)

APPENDIX 2. TURBOMAC EXPLANATION.

DIAGNOSTIC STATUS:

CRITICAL SPEED is PRESENT

COMMENTS:

Basically a design problem but often aggravated by poor balancing and poor foundation. Try to field balance rotor at operating speed, lower oil temperature, use larger and tighter bearings.

SYMPTOMS SUPPORTING THIS DIAGNOSIS ARE:
both the critical speed frequencies, rotor or stator resonant frequency and 1 x running frequency, are present (PRESENT)
amplitude of vibration is horizontal (moderate)
amplitude of vibration is vertical (M)
amplitude of vibration is on the shaft (moderate)
amplitude abnormally peaks as speed of machine increases (moderately strong)
loud roar sound (moderate)
bearings may have failed - wiped (weak)
bearings may have failed - fatigued (weak)

SYMPTOMS COUNTER TO THIS DIAGNOSIS ARE:
none

IMPORTANT UNREPORTED SYMPTOMS ARE:
seals rubbed (moderate)

IDENTIFIED POSSIBLE CAUSES OF THIS PROBLEM ARE:
may be excessive forces and moments in piping
expansion joints may not be properly installed
piping may not be properly supported

UNREPORTED POSSIBLE CAUSES OF THIS PROBLEM ARE:
none
none

APPENDIX 3. ACTUAL CASE HISTORY.

Everything was going well for the engineers at a midwest utility. They had just finished some minor work on a 750 horsepower boiler draft fan during a scheduled shutdown and the plant was ready to come back on line. The maintenance engineers had to be sure the fan worked before they could put a fire in the boiler. The engineers started it up.

As the fan came up to speed, the engineers noticed something was wrong. The fan didn't sound right and they could feel abnormal vibration coming through the floor slab. The machine had just been aligned. What could be wrong? How should they begin to find out?

The engineers called on TURBOMAC, the turbomachinery diagnostic expert system from The Hartford Steam Boiler Inspection and Insurance Company. They took some vibration readings with a hand-held meter and answered the questions about the problem. Within seconds, TURBOMAC suggested that the problem might be misalignment.

The machine had just been painstakingly aligned, so this possibility had been ruled out by the experts. They were convinced instead that it must be a balance problem. Solving this type of problem would require a long trial-and-error process of welding balance weights onto the shaft and re-running the fan until the vibration was corrected.

Since an alignment check was quick, the engineers decided to perform this test. To their surprise, TURBOMAC was right. The machine was out of alignment. They re-aligned the coupling and started the fan back up. The problem was solved. The unplanned outage of this fan could have cost the utility about $10,000 for each hour.

Metermen's Assistant Software (MAS): An Expert System Application at PG&E

EUGENE C. KONG
Pacific Gas and Electric Company
Electric Distribution Department
123 Mission Street
San Francisco, California 94106, USA

ABSTRACT

Pacific Gas and Electric Company (PG&E) has introduced expert systems into several aspects of company operations. This paper presents one application of expert systems called the Metermen's Assistant Software (MAS). It is designed to assist field personnel in servicing and maintaining of over four million of PG&E's electric revenue meters. MAS provides the ability to quickly index and reference information related to safety precautions, diagnostics, operating policies, testing and maintenance procedures.

The nature of this application lends itself nicely to the use of a rule based expert system. MAS is PC based and currently is piloted with a limited sized knowledge base. Since a detailed knowledge base in the area of metering can be quite large, it was the objective of the pilot to determine the areas of knowledge for expansion.

The details of the design and implementation are discussed in this paper. The knowledge base organization is also described. The impact from the introduction of this system is assessed and the reactions from the users are summarized. This paper concludes with some suggestions for future implementation and expansion of this system.

INTRODUCTION

The Metermen's Assistant Software (MAS) is an application of expert system which assists PG&E personnel in maintaining and operating electric revenue meters. To understand the purpose of MAS and its potential benefits, let us first examine how things have been done traditionally and contrast this with recent changes.

PG&E serves a territory of approximately 94,000 square miles in central and northern California. There are about four million electric revenue meters in the system. With a variety of customer classes (different voltages, current and wire configurations) and the accumulation of many vintages of metering equipment, there exist many different types and makes of meters at PG&E. Add to that the recent implementation of a number of complex rate tariffs and the more sophisticated multi-function meter registers, more demand is now placed on field maintenance personnel to know more. Metermen not only must be well versed with traditional knowhow to install and test electromechanical meters but also must acquire the new knowledge to deal with electronic microprocessor based meters.

There are approximately 160 metering personnel dedicated to maintaining the 4 million meters. Apprentice metermen are trained on the job as well as with some classroom instructions on the basics of testing and servicing electric meters. Apprentice Metermen progress to become Senior Metermen as they gain more field experience. There are other labor classifications that also assist in meter installation and maintenance but only to a limited degree based on their training and qualifications. These include Electric Troublemen and Gas Servicemen. Maintaining a high level of expertise in the field has been a challenging goal and has been made even more difficult with recent reorganization and pressure for down-sizing.

There are a host of references provided to the metermen for guidance. These include the Engineering Standards which detail the construction and installation, the Company Standard Practices which establishes the Company's official policies, Operating Bulletins which have procedures and guidance related to transmission and distribution, and more specific to metering is the Electric Meter Manual as well as many Intra-Company Memos as new situations arise. In short, there is an over abundance of information that the metermen must be aware of which was not covered under the traditional training program.

Accurate and timely dissemination of new information and operating procedures is essential to PG&E's daily operation. Amidst all of the available information, it is the objective of MAS to be the metermen's tool in sorting out important points and applicable information.

FUNCTIONAL DESCRIPTION

In concept, MAS is designed to provide the user with information as to what to do next, where to get additional help or reference, what is important and why. MAS first prompts the user with a series of questions to identify the situation and conditions at hand, then concludes with a list of applicable recommendations. Each recommendation is rated in its importance in order to distinguish the must do's from the nice to do's.

MAS provides assistance in the following metering tasks:

1. Servicing and Maintenance. Information relating to equipment recalls and special safety precautions will automatically be provided to the user. MAS uses qualitative parameters such as physical traits and markings to distinguish those meters under recall.

   Specific references are made to documents (when available) which contain step by step instructions.

   MAS eliminates unnecessary servicing costs by providing the guidance as to when it is economical to replace a defective meter with a new one. MAS also assists in deciding if its is worthwhile to return the defective meter to PG&E's central meter shop for repair.

   MAS notes as a rule of thumb which meter types are to be retired and also notes which parts are worth salvaging.

   MAS provides information on replaceable parts such as batteries or electronic chips.

   MAS will provide standard procedures in dealing with meter tamperings and energy thefts.

2. Testing. Testing and verification of meter installations vary depending on meter type. MAS can provide assistance in test setups and calculations.

3. Programming. To implement the various rate options available to its electric customers, PG&E uses meters with programmable registers. While there have been some progress made to standardize the programmable electric revenue meters in the industry, there still exist many differences from manufacturer to manufacturer. MAS provides specific programming information applicable to the meter type selected by the user. MAS also warns of common programming mistakes to avoid.

4. Troubleshooting. This is one of the most useful area served by MAS. MAS can pre-warn the user with all the past problems and solutions experienced by that meter type at hand. For example, a particular type of meter (manufactured during certain years, belonging to a particular serial number family, etc.) commonly experiences an unique type of failure. MAS can remind the metermen to check for such abnormality.

MAS can also warn the user of the common mistakes that are made which can cause those symptoms observed.

There are often cases where time is needlessly spent on troubleshooting because there was actually nothing wrong with the meter. MAS provides the user with common false alarms experienced from the past. For example, there was a case where one particular type of electronic meter was often reported as having a dim display. The problem turned out to be that when viewed with polarized sunglasses, the meter display can be dim or blank.

In the case of electronic registers with error codes showing, MAS can provide English translation of the problem.

5. Record Keeping. One of the successes and challenge in maintaining the 4 million meters at PG&E is the record keeping. PG&E has a large and complex database system called the Meter History System which keeps track of these meters. To insure the integrity of the data, MAS can issue special notes regarding the record keeping of each different type of meters.

Other design features of the MAS include the following:

- MAS is PC-compatible with the lap-top computers currently in use at PG&E for programming and interrogating electronic meter registers.

- MAS is self contained on a single 360K diskette which provides ease of transportability and distribution of the software through out PG&E's service territory.

- MAS can be used as a training tool. MAS can answer the question "WHY?" as well as displaying documentary files.

- MAS calls external graphics programs for additional clarification when words have difficulty describing the situation.

IMPLEMENTATION

MAS is implemented using the PC version of a rule-based expert system shell call EXSYS. EXSYS is a trademark of Exsys, Inc.

Currently, the MAS pilot is implemented using approximately 75 rules. MAS is configured for backward chaining so that all applicable rules are used to obtain the final recommendations. Rules are written in if-then-else format. These rules are based on all the applicable written Transmission and Distribution Bulletins, Standard Practices and Metering M&O Memos as well as the author's personal expertise. Usually a single memo or bulletin can be represented by one rule.

The following is an example of how one such rule is represented:

==

IF:
The type of meter examined is -- Meter with Electronic Register
and The register type is -- MTR-20

THEN:
Check meter battery. - Probability=9/10
and Check meter programming. - Probability=9/10

NOTE:
There are some batteries made by SAFT which have a small plastic label on the battery contact tab that interferes with the battery carry-over function. All updated registers with the new daylight savings time have chips that have a yellow coloring painted on the edge. All meters should be checked for the latest program chip.

REFERENCE:
System Metering M&O Memo #25

==

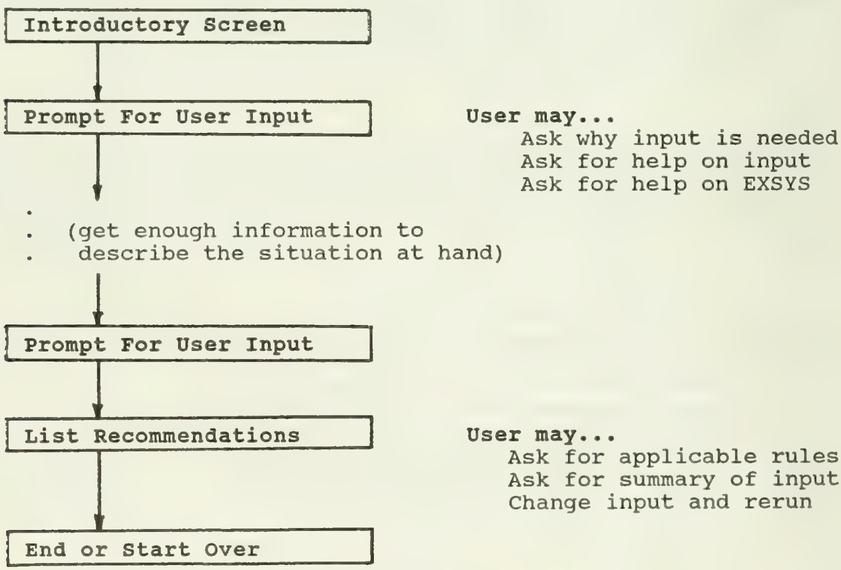
The preceding example illustrates many aspects of the MAS implementation. MAS uses natural English expressions and descriptions. This allows the rule to be read directly and is easily understood by the user when MAS is used as a training tool or when the user is backtracking to examine how the final recommendations are arrived at. The rule can also be read to tell the user why certain questions are being asked by MAS.

With this format, future modifications and editing can be easily done. EXSYS comes with a utility for maintenance of the MAS rules. Editing capability is not made available to the MAS users.

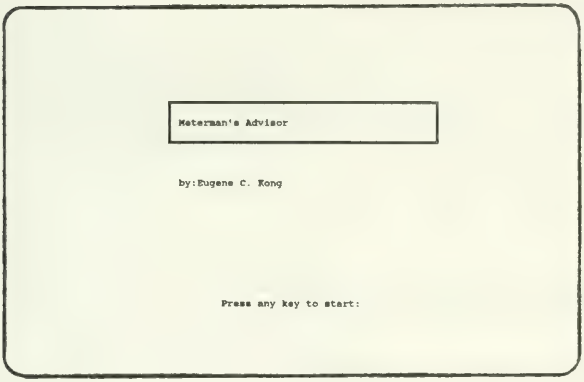
When the user asks why, the applicable rules are then displayed. The NOTE and REFERENCE section gives the user additional information relating to the origin of the rule. The REFERENCE section is used to point the user as to where to obtain additional information or to cite an official Company policy.

MAS uses the probability value assignment to rank how important the recommended actions are. The probability value system used by MAS is a scale of 0 to 10. Those ranked 8, 9 or 10 are "must do's".

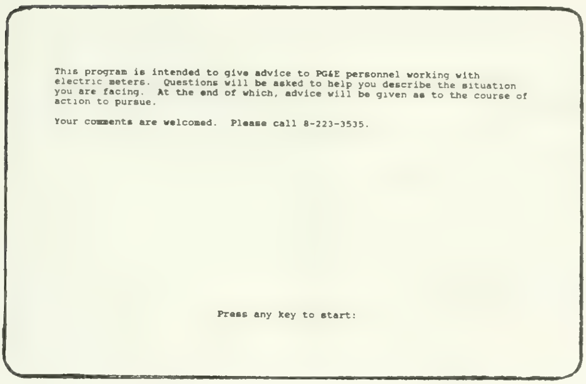
The following flow chart shows how a typical session may progress when using MAS.



The following are examples of the interactive screens as seen from the user's point of view:



These are brief introductory screens.



```
The meter has
     1   mechanical dials present.
     2   no mechanical dials.
     3   its mechanical dials already removed.
why

Enter number(s) of value(s), WHY for information on the rule,
<?> for more details, QUIT to save data entered  or <H> for help
```

This is a sample of an input screen. Here the user may ask why the information is needed.

```
RULE NUMBER: 28
IF:

     (1)    The type of meter examined is -- Meter with Electronic Register
and  (2)    The meter has  mechanical dials present.
and  (3)    The REGISTER Manufacturer is -- NOT Sangamo
and  (4)    The Sangamo REGISTER type is -- NOT MTR-20 (MTM-20)

THEN:

            Remove Mechanical Dials. - Probability=9/10

NOTE: Meters with redundant mechanical and electronic KWH readings should have
      the mechanical dials removed.

IF line # for derivation, <K>-known data, <C>-choices , <R>-reference,
 or  - prev. or next rule, <J>-jump, <H>-help or <ENTER> to continue:
```

In answering why, the rule being tested is displayed to the user.

The following screen will display a list of advices to consider. These advices have associated values from 0 through 10. The higher the value means that the more important it is to perform this action. Those ranked 8, 9, or 10 are "must do's".

If you strongly disagree with the advice given or if you believe there is an error, please notify Eugene C. Kong at 8-223-3535. Your comments are welcomed and appreciated. It will also help in developing a better tool for all metermen.

Press any key to display results:

After sufficient input is obtained, MAS displays its conclusions and recommended actions.

User feedback is encouraged to help keep the knowledge base updated and consistent.

	Values based on 0 - 10 system	VALUE
1	Remember to use RUBBER GLOVES.	10
2	Install/Check Voltage Stabilizer.	10
3	Install/Check Meter Protective Device (Lightning Arresters).	10
4	Check meter programming.	9
5	Write program info on meter.	9
6	Check interference to the disk rotation.	9
7	The service voltage (volts) = 480	
8	The Manufacturer's Meter Serial Number = 999999999	

All choices <A>, only if value>1 <G>, Print <P>, Change and rerun <C>, rules used <line number>, Quit/save <Q>, Help <H>, Done <D>:

```
RULE NUMBER: 56
IF:

     (1)     The type of meter examined is -- Meter with Electronic Register
and  (2)     The REGISTER Manufacturer is -- General Electric

THEN:

             Check interference to the disk rotation. - Probability=9/10

NOTE: This meter has a wiring cable which may interfere with the disk movement.
       Care should be exercised when mounting the cover to avoid this problem.



IF line # for derivation, <K>-known data, <C>-choices , <R>-reference,
 or  - prev. or next rule, <J>-jump, <H>-help or <ENTER> to continue:
```

This Screen demonstrates the retracing of how conclusion #6 was arrived.

```
1    The reason for investigation is -- Office Request
2    The type of meter examined is -- Meter with Electronic Register
3    Upon initial inspection, the meter .. is in good physical condition.
4    The electronic display is -- Showing An Error Code
5    The error code showing is -- Er000010
6    The meter has  no mechanical dials.
7    The type of customer is -- Agricultural
8    The meter is -- Polyphase
9    The Dial Constant is -- one.
10   The area of assistance needed is -- General Advice
11   The REGISTER Manufacturer is -- General Electric
12   The G.E. Register Type is -- TM-91
13   Variable [SERVICE VOLTAGE] = 480.000000
14   Variable [METER SERIAL] = 999999999.000000



Enter number of line for additional information,
or <ENTER> to continue:
```

This screen summarizes the user's inputs.

USER REACTIONS AND IMPACT ASSESSMENT

The impact of MAS to PG&E's metering operations is unnoticeable at this time. The pilot version of MAS was introduced in early 1989. Initial reactions from the potential users were mixed. Some individuals thought that this would be a very useful tool in their daily work while some have very little comment. The major concerns were that Metermen may become overly dependent on the expert system which could lead to eventual decline in metering personnels expertise and that too much time might be spent on using the computer rather than performing field work. However, it was recognized that it would be difficult for any individual to keep in his head as much information as the potential knowledge content of MAS. In all, the reactions bear a great resemblance to the introduction of the lap-top computer to PG&E's field metering personnel. Some were for it and some were against it.

In spite of the differences in opinions expressed by the potential users, there were agreements to some points.

1. MAS would be useful in cases where the field personnel are faced with complicated tasks that are done only once in a while. For example, MAS should concentrate on providing the expertise to test a particular meter type which a meterman might encounter once a month and which requires some complicated procedure.

2. MAS should place less emphasis on metering tasks that are done routinely.

3. MAS is useful to supervisors of metering related personnel when supervisors are managing multiple discipline areas, i.e. gas and electric.

4. MAS is useful in providing uniform interpretation of rules and policies.

5. MAS serves as a good reminder for any equipment recalls. This is especially true for those equipment with multiple types of recalls.

6. MAS can provide useful support to those Company locations where there is a shortage of meter specialists.

7. MAS will become even more necessary as metering equipment becomes more complex.

CONCLUSIONS

The MAS pilot was a necessary step in the development of the expert system application in the area of metering at PG&E. The pilot was successful in identifying the areas of knowledge to include in the software as well as finding a format comfortable to the users. There are many recognized potential benefits, but to capture them requires that the users be comfortable with MAS and use it on a regular basis.

The MAS knowledge base is expected to continue to grow in size. Reorganization of the rules is necessary at an early stage to accommodate this growth. The MAS pilot has a broad knowledge base which includes the troubleshooting guide of many meter types together. Although it had the benefit of having a lot of knowledge at the fingertip, the system also tends to ask too many questions from the user's point of view. Breaking the rules into major meter type categories will help to minimize the number of questions asked by MAS.

The expert system shell, EXSYS, used by MAS is also capable of calling external program routines. This is a powerful feature that will be explored by future versions of MAS. The planned functions include the following:

a. Control and interrogation of automated meter test equipment.

b. External data retrieval to minimize user input.

c. Integration of MAS to other existing metering software.

d. Perform meter accuracy calculations.

To minimize the development efforts of MAS knowledge base by PG&E, manufacturers of new metering equipment will be encouraged to provide their troubleshooting information in the MAS knowledge base format as well as the traditional written instruction manuals.

Even though its initial acceptance is a little slow, the use of MAS is anticipated to grow rapidly Company-wide as it gains more exposure.

Safety Significance Evaluation System

BARCLAY S. LEW and DIXON YEE
Pacific Gas & Electric Co.
Nuclear Regulatory Affairs
333 Market Street
San Francisco, California 94106, USA

WARREN K. BREWER and PAUL J. QUATTRO
ABZ, Incorporated
Suite 150
1175 Herndon Parkway
Herndon, Virginia 22070, USA

KENNETH D. KIRBY
Science Applications International Corp.
Suite 360
311 Park Place Blvd.
Clearwater, Florida 33519, USA

ABSTRACT

The Pacific Gas and Electric Company (PG&E), in cooperation with ABZ, Incorporated and Science Applications International Corporation (SAIC), investigated the use of artificial intelligence-based programming techniques to assist utility personnel in regulatory compliance problems. The result of this investigation is that artificial intelligence-based programming techniques can successfully be applied to this problem. To demonstrate this, a general methodology was developed and several prototype systems based on this methodology were developed. The prototypes address U.S. Nuclear Regulatory Commission (NRC) event reportability requirements, technical specification compliance based on plant equipment status, and quality assurance assistance. This collection of prototype modules is named the safety significance evaluation system. The event reportability module was developed into a full-scale system to demonstrate that the methodology is robust enough to extend beyond the prototype state. Thus, the result of this effort is a general set of tools that can be easily adapted to other applications. Further, rapid prototyping can be accomplished for use in establishing system functional characteristics and then scaled-up to a full-site system. Some other possible applications of the tools developed are 10CFR50.59 evaluations, equipment tagouts, and operator shift scheduling.

Introduction

Operation of a nuclear power plant in today's regulatory environment involves compliance with a large number of rules. These rules are imposed on the utilities via the Code of Federal Regulations, the plant's technical specifications, and other site-specific procedures. In addition to the number of rules, the problem can be compounded when individual rules are not self contained and independent. Each day operators and other utility personnel are challenged to interpret the requirements and to act on their interpretations. Moreover, they are required to consider all the potential interrelationships of the rules without failing to consider any of them. The sheer volume of information and complexity makes this a difficult task even for the best personnel. In an effort to make this situation more manageable for utility personnel, the PG&E, in cooperation with ABZ, Incorporated and SAIC, investigated the use of artificial intelligence-based programming techniques to assist in regulatory compliance problems.

The result of this investigation is that artificial intelligence-based programming techniques can be successfully applied to this problem. A methodology showing this has been developed. Further, based on this methodology, prototypes have been constructed to handle some of the more complex regulatory compliance problems. This collection of prototypes is called the safety significance evaluation system. The prototypes evaluate reportability in accordance with U.S. Nuclear Regulatory Commission (NRC) requirements, technical specification compliance based on plant status information, and quality assurance requirements.

Regulatory Requirements and the Need for a Computerized System

This investigation focused on the requirements specified in the Code of Federal Regulations, plant technical specifications, and other plant-specific procedures. The requirements specified by these documents fall into two general categories. The first is where the information needed to determine compliance with the requirement is self-contained and does not depend on compliance or non-compliance with other requirements. The second category involves situations where multiple levels of information are needed to determine compliance with the requirement. Included in this category are situations where the status of compliance with other requirements is necessary. Two examples are provided to illustrate the two categories of requirements.

An example of the single level requirement can be shown by reviewing the Code of Federal Regulations, Part 10, Section 50.72.b(ii) which states:

> "the licensee shall notify the NRC as soon as practical and in all cases within one hour of the occurrence of any of the following:
>
> (ii) Any event or condition during operation that results in the condition of the nuclear power plant, including its principal safety barriers, being seriously degraded; or results in the nuclear power plant being:
>
> (A) In an unanalyzed condition that significantly compromises plant safety;
>
> (B) In a condition that is outside the design basis of the plant; or
>
> (C) In a condition not covered by the plant's operating and emergency procedures."

In this requirement, if the conditions described in either parts A, B, or C are satisfied, then the NRC must be notified within one hour of the occurrence. Note that each condition can be satisfied without knowledge of the other two. Thus, if the plant is in a condition outside the design basis of the plant, the NRC must be notified regardless of whether it is an unanalyzed condition or a condition not covered by the operating and emergency procedures.

An obvious example of a multiple level requirement is a plant's technical specification requirements for operability of Emergency Core Cooling Systems (ECCS). The requirements for a particular system (e.g., core spray) cannot be considered in isolation from the rules for the other ECCS systems. Continued operation with certain core spray system components inoperable is acceptable at least for a short duration, but only if the requirements for operability of

other ECCS systems are satisfied. Note that in this multiple level requirement, some of the requirements are of the single level type such as that discussed above.

In these examples, the domain of the requirement has been well defined. The result is clear. However, as the number of requirements increases and interdependencies of the multiple level rules become more numerous or complex, the burden on operators and other utility personnel to address all aspects of these requirements becomes large. In some cases, decisions about requirements are based on one's interpretation of the requirement. Thus, discrepancies begin to develop due to differences in opinion as to the meaning of the requirement. In other cases, the volume of requirements existing on a certain area creates the potential for one to forget or overlook one of the requirements. Based on these types of concerns, PG&E, ABZ, and SAIC developed the methodology and prototype systems discussed in this report.

It is clear that a computerized system could alleviate the concerns discussed above. For instance, by developing rules for a computerized system, the utility's interpretation of the requirement would become an inherent part of the system. Thus, when different operators use the system they have the benefit of the entire operations and engineering staff's interpretation in the system and at their fingertips. Further, if a large number of requirements exist, they are kept in the computer system. This means when the system is queried, the computer ensures that all the requirements are addressed -- not just those which the operator remembers. In short, a computerized system ensures that all evaluations are complete and consistent.

General Methodology

The general methodology underlying the safety significance evaluation system is an easily adaptable, interactive expert system. However, as will be discussed later, the interactive nature has not been maintained in all prototype applications. As depicted in Figure 1, the system consists of three main parts: the rule base, the knowledge base, and the inference engine.

The rule base is, as in any expert system, the collection of the known rules governing the phenomenon or process for which the system will provide expert advice. The safety significance evaluation system allows these rules to be represented as a set of questions with required responses or as a set of logic trees. This distinction in how the rule base is represented is purely cosmetic, in that it changes how the rules appear to the system user, but makes absolutely no difference in how the rules are represented, stored, or manipulated within the program. In general, the rule base has been represented as a set of questions and answers for these prototype applications that are interactive in nature. The logic tree representation has been chosen for applications that are not interactive. This will be explained in greater detail in the section of this report dealing with details of the prototype applications.

The description of the system as easily adaptable is intended to portray two characteristics. First, the system allows the user, with no programming knowledge and very little training in the system, to modify the rule base. The user can accomplish this by simply filling in blank data fields to describe the rules. The system then converts this input into a form used by the computer. Second, the underlying programming has been maintained general enough to allow the system to be quickly reconfigured or modified to be applied to a wide range of applications. This should be clear from later discussions of the three different prototype applications investigated.

The knowledge base is the collection of facts or data which describes the event, situation, or item to be evaluated. The knowledge base, like the rule base, can be represented in two ways. However, in the case of the knowledge base, the distinction is substantive. The first way of representing the knowledge base is as the set of interactive responses obtained from the user at run time. The system has no knowledge or facts other than the responses provided by the user to the system.

The second way of representing the knowledge base is a stored database. The database is updated as needed by the user. When the system is run to provide evaluation or analysis, the system utilizes the stored data. There would be interaction with the user to obtain input only where the database does not contain sufficient information to allow the system to reach a conclusion.

Finally, the inference engine is the part of the system that combines the rule base and knowledge base. The inference engine applies the rules to the known facts to arrive at conclusions or provide advice. An important part of the inference engine used in the safety significance evaluation system is the generation of the "why" reasoning for a conclusion. That is, as the system processes information to arrive at a conclusion, the reasons for reaching that conclusion are "remembered." The reasons for the conclusion are provided to the user along with the conclusion. This is important to allow users to determine if they agree with the system's conclusion.

The inference engine also provides the user with the required actions based on the conclusion. The required actions are part of the user input rules and, thus, can readily be modified as required by the user with no programming effort.

Applications of the Methodology

Based on the framework of the general methodology, PG&E, ABZ, and SAIC developed a prototype system named the safety significance evaluation system. The system consists of three basic applications:

1. Event Reportability Module

2. Limiting Conditions for Operation Module

3. Quality Evaluation Module

The applications operate independently or as an integrated system. Since the start of development on this project, the Event Reportability Module has become more than a prototype and is considered to be an actual operating system capable of handling all aspects of the NRC requirements for event reportability. The other two modules are still in the prototype stage and would require further development to handle large-scale problems.

The Event Reportability Module is a completely interactive, menu-driven system with all the functions necessary to assist utility personnel with NRC event reportability requirements. The system does not require the user to have any knowledge of the underlying computer language. Utilities are built into the system which allow the user to easily modify the contents of the system. The system consists of several modules including a module for system operation, a module for writing and changing the rules, a module for performing reportability evaluations, and other modules for utility functions such as providing printouts.

The Event Reportability Module requires a rule base before any evaluations concerning event reportability can be performed. Therefore, the user must develop this rule base using the system utility for writing such rules. In this case, the user must enter a series of questions and answers defining the reportability rules as interpreted by the user utility. Figure 2 is a screen image from the rule entry utility. A logic statement is then written that links the series of questions. In an evaluation session, the user is asked a series of questions from the rule base regarding the specific event under consideration. The user responds to the questions with a "YES" or "NO" answer. Figure 3 is an example of the screen image during the evaluation session. The inference engine then evaluates the responses to determine whether a report is required. Figure 4 shows the results of the evaluation. This evaluation is based upon the user's interpretation of the reportability requirements as input in the rule base.

There are several aspects of this system that make it especially user-friendly. First, the system rule writer allows one to enter question-specific help. Thus, if a question requires information from a table, the table can be included in the help. More importantly, if a question is vague, it can be well-defined in the help as to what it means for the specific plant. The rule writer also contains fields to enter information about the type of report required such as one-hour notification or a licensee event report as well as when these reports are due. The system also maintains a log of the session and explains why the event is reportable. Finally, all this information is available on hardcopy if desired.

The functions of the Limiting Conditions for Operation Module (LCOM) are more sophisticated than those of the Event Reportability Module but use the same underlying principles and techniques. This prototype is a demonstration of these functions and is not intended to be a full-size system in its present configuration. The two primary differences between LCOM and the Event Reportability Module are that LCOM can handle situations involving multiple level rules and that a plant status database is maintained by the system. The addition of these capabilities results in a powerful system as described below.

LCOM is a menu-driven system with the functions necessary to assist operations, maintenance, and engineering personnel in complying with technical specification requirements. In this module, the program utility for generating the rule base allows the input of upper level rules and lower level rules. The utility allows the upper level rules to be linked to the lower level rules. The lower level rules are generated independently. The result of this process is a logic tree configuration. An example using the technical specification section regarding the ECCS system will clarify the two types of rules. An upper level rule would state that, for compliance with the technical specification section for ECCS, the systems which constitute ECCS such as the high pressure core spray and low pressure core spray systems must be operable. The lower level rules would be the conditions that make core spray systems operable, such as pump or valve operability and availability of instrumentation. Additional rules could be written to describe the conditions for pump operability such as power supply availability and lube oil availability. The rules can go to the level of detail desired by the user.

The system also maintains a database of the status of plant components such as whether valves are open or closed or whether pumps are operable or inoperable. Database access is provided via a graphics display of a schematic of each plant system with a mouse. A database interface has been developed for the LCOM application which allows displaying the database information by use of a system schematic. A simplified schematic is shown in Figure 5. The user simply

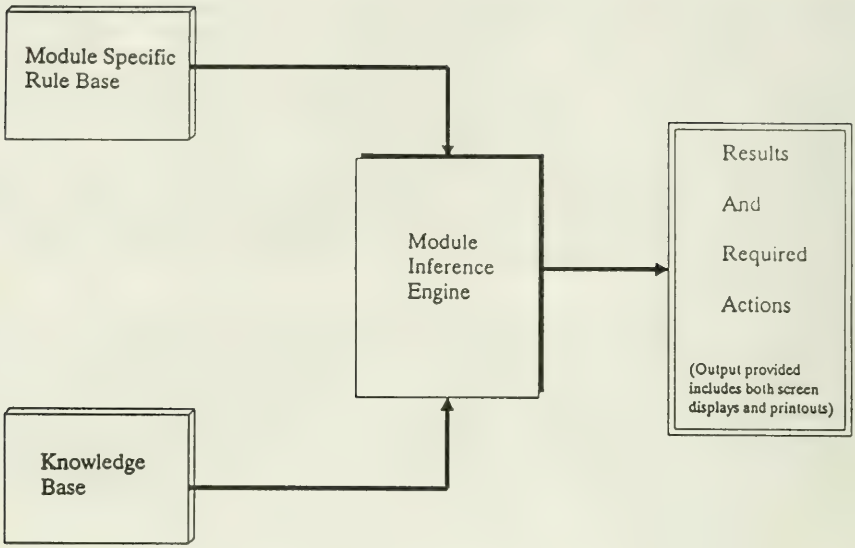


Figure 1. System Module Operation

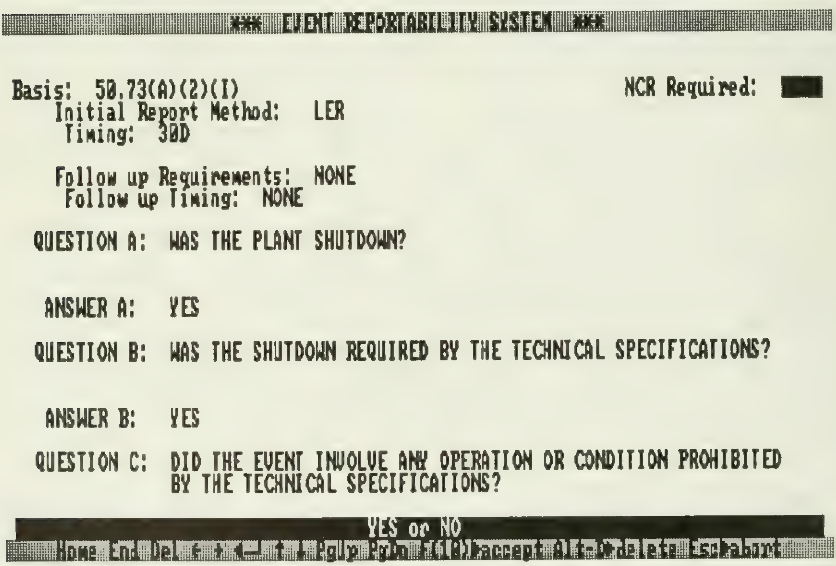


Figure 2. Event Reportability System Rule Writing Utility

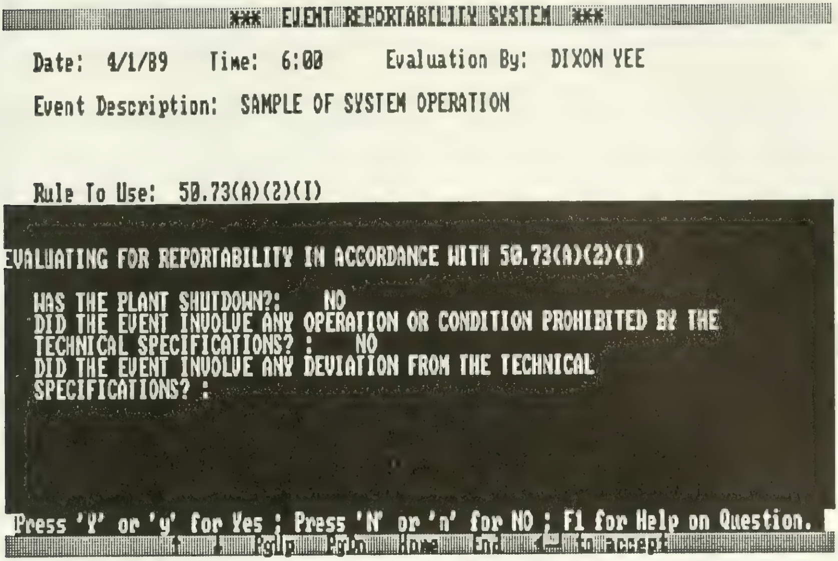


Figure 3. Event Reportability System Display of Results

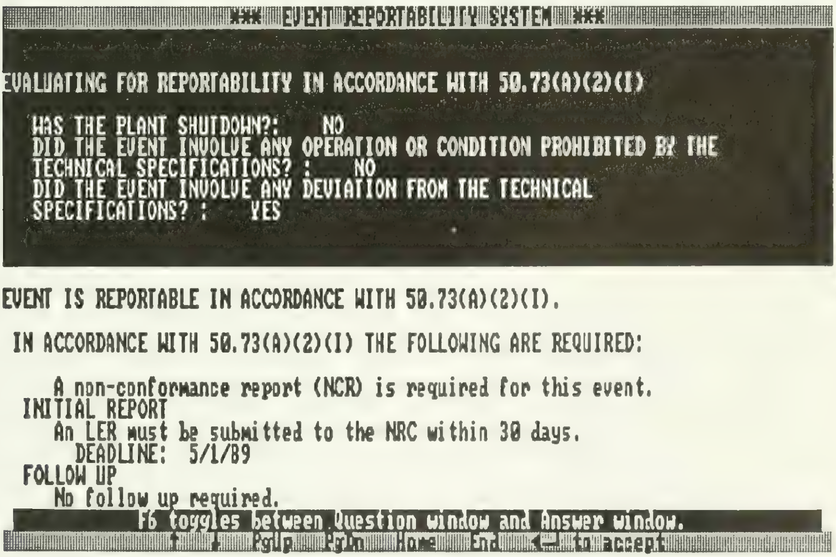


Figure 4. Event Reportability System Interactive Evaluation Session

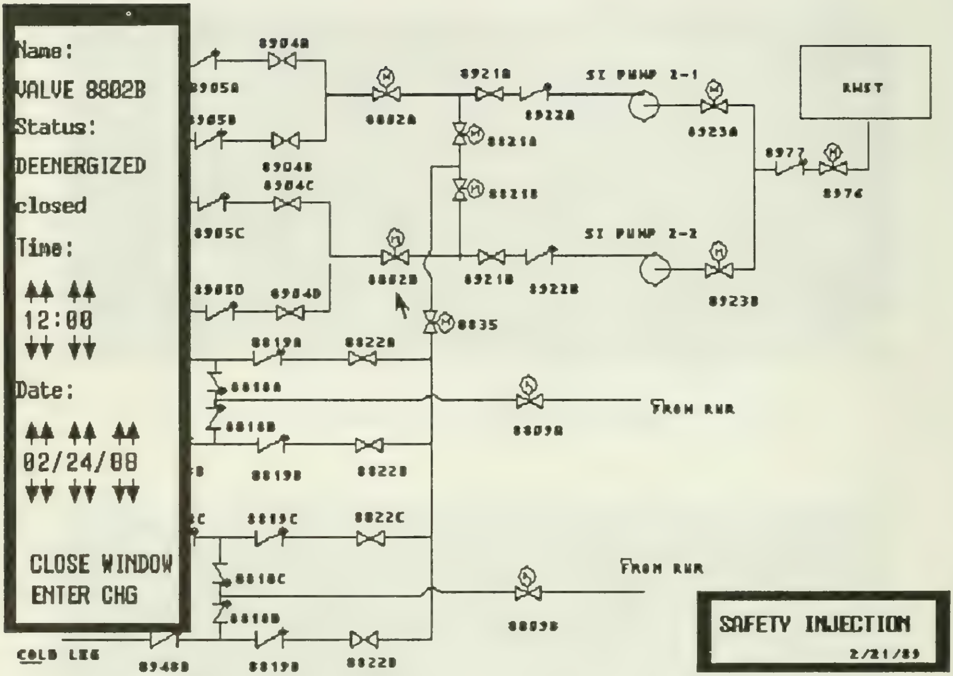


Figure 5. Limiting Conditions for Operation System Plant Status Database Facility

points with a mouse at the component for which he wishes to update the information and clicks the mouse button to select the component. The current information for that component is then displayed and the data may be updated using the mouse and clicking through the possible states.

After updating the database, the system will analyze the information and make a determination as to whether the plant is in compliance with each section of the technical specifications. If the plant is not in compliance with the technical specifications, an explanation as to why it is not is provided. LCOM also will provide a hardcopy of the session.

The final prototype system is the Quality Evaluation Module. This module provides plant personnel with assistance in determining whether a quality evaluation is required. The system functions in exactly the same way as the Event Reportability Module with respect to rule preparation, interactive operation, an explanation as to why a quality evaluation is required, and a hardcopy of the session. This module has one added feature in that a component Q-list is contained in it such that the system can automatically determine whether quality-related equipment is involved.

Details of Prototype System

The prototype system and applications are programmed in the artificial intelligence language PROLOG. The use of PROLOG rather than an expert system shell was chosen to avoid the pre-defined limitations inherent in any shell and to allow the development of a customized "shell" which could be readily fine-tuned for specific applications.

The computer code is highly modularized which provides two specific advantages. First, this allows easier modification of the code. The applicable program section can be more easily located and modifications can be tested more simply as an isolated module before testing as part of the entire system. Second, modular construction provides for rapid prototyping of new applications. The program modules provide basic building blocks which can be modified and rearranged to match new applications. The remainder of this section is a more detailed look at some of the program modules.

The first module is the rule entry/modification module. This is the module that provides the user interface that allows non-programmers to build, update, and maintain the required rule base for an application. This module allows the user to fill in data fields on the screen and thus modify, add, or delete rules from the rule base.

The rules are generally of the form that a specified condition is true if some specified set of conditions is satisfied. In adding such a rule to the rule base, the necessary conditions are simply listed without regard for the logical connections among these conditions. When all the necessary conditions have been listed, the user must then input how these conditions are logically connected. Take for example, a rule that has four conditions, A, B, C, and D, which are to be part of the rule. The logical connections among these conditions are then specified by writing a logic statement such as "A AND B AND C AND D." This is the simplest logic statement that can be written for four conditions, but the rule module allows the logic to be as complex as needed to represent the desired rule. The logic can be made more complex by the use of "OR" as well as "AND" and by the use of parentheses to define groupings. In general, parentheses are required only as necessary to ensure the statement is unambiguous.

For example, again using four conditions, the logical statement, "A AND (B OR (C AND D))" could be written. This states that the rule is true if A and B are satisfied or if A and C and D are satisfied. Using combinations of "AND" and "OR" and parentheses, the logic statements written and hence the system rules can be arbitrarily complex. The system user must only be able to write an unambiguous logic statement describing the rule.

The system can utilize arbitrarily complex rules is a very important difference between the safety significance evaluation system and expert system shells. Allowing the rules to be arbitrarily complex lets the user write the rules as he knows or thinks of them. The user does not have to alter the rule by breaking it into more than one rule in order to conform to program limitations. Most expert system shells have limits on the complexity of the rules that can be entered but provide means by which the essence of a very complex rule can be input by entering two or more less complex rules.

The next module is the logic evaluation module. This is in essence the inference engine. The module provides the mechanism for applying the arbitrarily complex rules developed through the rule module to the data from the knowledge base to reach some conclusion. This module provides several options as to how the rules are applied.

First, depending on the application or at the discretion of the user, the inference engine may apply the rules in an exhaustive fashion. That is, the system would apply <u>all</u> the rules in an attempt to determine <u>all</u> the reasons for the arrived at conclusion. Alternatively, the system may only apply those rules necessary to reach conclusion. In this case, the system would be able to indicate only a single reason for reaching the conclusion.

Next, the ability has been provided for a rule to refer to another rule. That is, one or more of the conditions listed in a rule may be another rule. Although the capability to perform this evaluation is embedded in the logic evaluation module, there is no limitation in the rule entry module that prevents listing a rule as one of the conditions in a second rule.

This layering of rules, that is having a rule refer to another rule, in theory has no limit to the number of layers. The actual limit is, therefore, determined by the memory capacity of the host computer.

The solution to problems becoming too large for PCs is to move to a larger, more powerful computer. The particular version of PROLOG used in the prototype system is available for DEC VAX machines. Also, converting the system to a version of PROLOG compatible with any UNIX-based system appears to pose no significant problems.

The final module to be discussed is the database module. This module is not used in applications which are completely interactive such as the event reportability application. However, it is very important in applications such as the prototype LCOM that depends on a stored database. The database module, as well as the rule module, contains the coding to provide for protection from unauthorized changes to the data. Specifically, the system includes multiple-level password protection which can be fine-tuned to the user's desires. That is, different program functions or data can be assigned different required password levels. Thus, very close control of the use of certain program functions is achieved while more general use of other functions is allowed.

Also, the database module keeps a log of all database changes made and who made the changes. This logic can be used as a means to ensure that no incorrect

changes are made and to determine when and by whom incorrect information was entered if it is determined that incorrect information has been entered. This would be important to allow rerunning any evaluations done utilizing the errant data to determine if there is any change in the conclusion.

The database module has also been provided with the necessary open ends to allow more than one database. Specifically, the system could accommodate a real database as well as one or more hypothetical databases. The real database would be subject to password protection and logging as discussed above, but the hypothetical databases would not. The hypothetical databases would be used for analyzing "what-if" questions without the danger of corrupting the real database.

Conclusions

The safety significance evaluation system prototype applications have demonstrated the ability to apply expert system technology to a range of problems facing utilities that operate nuclear power plants.

Also, this work has developed a set of tools that can be readily used to extend this technology to other specific applications. These tools would allow for quickly prototyping a new application. Such a prototype, just as the prototypes developed thus far, would be useful in determining the desirability of such a system, establishing the system capabilities, and allowing the detailed system specifications to be determined after significant hands-on experience with a working prototype. This will result in more cost efficient and timely development of finalized applications. Also, the prototypes developed to date provide a catalog of ideas and features which can be viewed in operation and readily incorporated into other prototypes or finalized systems.

Some other possible applications of the tools developed are: (1) 10CFR50.59 evaluations; (2) equipment tagouts; (3) operator overtime control; and (4) operator shift scheduling. The tools developed are generally usable for any rule based application and thus there are certainly many other potential applications.

Also, the event reportability system has progressed beyond the prototype state. The only work necessary to convert this to a working system is for a user to define in finer detail the rule base. The features to allow this are already in the program and thus it is simply a matter of the user setting down in detail the interpretation of the event reportability rules.

The event reportability system, with the existing somewhat coarse rule base, has been tested against a sample of events described in actual LERs from Diablo Canyon and been shown to agree with the reportability determinations made by PG&E personnel. Furthermore, the logic processing module of the event reportability system has been extensively tested and verified to apply the rules input by the user. Thus, testing to verify operation with a different rule base is really a check that the rule base was input correctly.

A Causal Qualitative Modeling Approach Applied to Plant Disturbance Analysis

M. GARAKANI, B. FROGNER, D. KUHLMAN, and M. MILLER
Expert-EASE Systems
1301 Shoreway Road
Belmont, California 94002, USA

S. GUARRO
Lawrence Livermore Laboratory
Livermore, California 94550, USA

ABSTRACT

This paper describes the development of a system based on a Time Flowgraph Methodology (TFM) that can be used to perform plant disturbance diagnostic analysis. The specific application that is being developed has two major objectives: 1) assist plant operators in the early identification and analysis of disturbances that may lead to trips, and 2) when a trip occurs, assist in diagnosing the root-cause more quickly, thus allowing a faster recovery with less downtime.

This paper is mainly focused on describing the basic concepts associated with TFM. This new approach is based on incorporating the element of time and the concept of fuzziness in a causal qualitative model of the disturbances in the plant.

The resulting software is being integrated into EASE+ which is a tool-kit for development of graphical user-interfaces for process monitoring. This will facilitate the development and use of the TFM models to the point where a large segment of the industry potentially may use it. Furthermore, since EASE+ already has an interface to an expert system for processing of conventional production rules, it now has the potential of solving problems involving a mixture of causal and heuristic reasoning. The initial prototype is being developed on a PC/386 workstation.

INTRODUCTION

Symptom-Based Versus Causal Modeling Reasoning

Most expert systems used for diagnosis base their decisions on a symptom-based or "shallow reasoning" approach. In such an approach, the heuristic rules representing the symptom-based reasoning do typically not reflect the process topology; i.e., there is no clear mapping between the objects of the domain and the diagnostic rules [1]. This has some basic shortcomings:

- o Modifications of the plant configuration can be difficult to incorporate into the knowledge base since small changes in the process may affect a large number of rules. This makes the knowledge base difficult to maintain.

- o Rule-based systems are not easily portable as they make no distinction between plant-specific knowledge and general knowledge.

- o In a newly started plant, there is typically a lack of heuristic knowledge for the correct identification of faults. This is also the case for faults that are so rare as not to have provided the plant personnel with the needed experience.

- o Often the propagation of the original disturbances leads to a large number of secondary disturbances. This makes the task of going from the observed symptoms to the original fault difficult since the direct symptoms of the fault will be hidden among a potentially large number of secondary effects.

Because of these shortcomings, the symptom-based approach is most appropriate in problem domains where the fundamental knowledge of the causal mechanisms of the system is lacking. A commonly mentioned example is that of medical diagnosis where the exact causal pathways leading to the disease may not be known.

In contrast to the situation in the medical domain, the physical principles underlying a process plant are relatively well understood. Propagation of disturbances in a power plant usually involves cause and effect relationships that can be represented by relatively simple models.

Propagation of Disturbances

Generally a transient in a plant begins when one or more variables become disturbed. This is the start-up phase of the disturbance in which an initial fault such as a broken pipe or clogged valve causes initial disturbances in a number of plant parameters. This leads to the second phase of the transient during which the initial disturbances propagate throughout the system, each event causing others, leading to a whole shower of events occurring at various locations and times.

The original fault is known as the "root cause". The initial disturbance resulting from the root cause can be viewed as the fingerprint

of the root cause. This fingerprint might consist of a single event, or it may involve a larger number of disturbances which typically appear at a close proximity (both in space and in time) to one another. Propagation of these fingerprint events leads to a number of secondary disturbances, some of which might be of large enough magnitude to induce additional faults (with their own fingerprints) in other locations in the plant.

Due to the large number of events that may occur during a major plant upset, the task of identifying the root cause may be difficult. Post-trip analysis is an example where finding the original cause may require a substantial amount of work. This involves sorting out a large number of events, finding those which lead to other events, and also identifying any new faults which have occurred as a result of the propagation of the original events.

Although the time sequencing information is useful, there are many occasions in which it is not sufficient. All causal relationships involve a time interval between the occurrence of the cause and the appearance of its effects. Using this time constraint information can be very important in many diagnostic situations. Similarly, plant disturbances in one location can be causally related to other earlier disturbances only if they obey certain temporal constraints.

People frequently rely heavily on this other type of temporal constraint knowledge when reasoning about causal phenomena. There is a notion that the effect must follow the cause in some sort of qualitative time envelope. Should the effect appear at a time substantially different from when it is expected, it is generally ruled out as the consequence of the cause under consideration.

This suggests that a qualitative treatment of temporal phenomena is an important issue in process diagnosis. The time information, however, is only one of the issues that can benefit from a qualitative treatment. In addition, the approach described in this paper also includes the qualitative representation of parameter perturbations, transitions and causality.

Our goal is to create a methodology for process diagnosis that can support the qualitative modeling of time-dependent causal phenomena in process plants. It is true that quantitative simulation provides more information, but this is accomplished at a much higher computing cost, it requires extensive knowledge of parameter states and histories, and it is not applicable when a theoretical and rigorous understanding of the phenomena is lacking (or hard to provide). Our qualitative approach, however, would be able to make useful diagnostic and recovery recommendations based on a much less rigorous understanding of the plant's state and behavior.

LOGICAL FLOWGRAPH METHODOLOGY (LFM)

LFM is a methodology used for modeling the causal propagation of parameter perturbations in process plants. It was originally conceived and developed by Sergio Guarro and David Okrent [2,3].

In LFM, perturbations in the values of plant parameters are represented qualitatively by a small set of discrete values. LFM uses a set of five discrete states: "normal", "small positive disturbance", "large positive disturbance", "small negative disturbance", and "large negative disturbance" which are respectively represented by 0, +1, +10, -1, and -10.

Parameter perturbations in one location can have a causal influence on other parameters. The causal relationships being modeled are those of the physical constraints of the normal or abnormal process, including those dictated by the process control logic. An example would be the causal relationship between the neutron flux and the reactor core temperature; e.g., an increase in neutron flux increases the reactor core temperature.

An important feature of LFM is that it provides specific mechanisms for specifying how causal relationships can change under the presence of faults or other process conditions. This is accomplished by the operation of what is known as the conditional network, which is formed by test boxes and boolean elements (AND and OR gates).

In disturbance analysis applications, the LFM model is traced on-line to generate all possibilities that can potentially induce a disturbance in a given plant parameter. The result is presented in the form of a fault tree. The fault-tree contains all of the explanations for the top event which are theoretically possible. However, not all of the explanations contained in the fault-tree are consistent with the observed plant measurements. Therefore, the fault-tree is matched with the process signals to generate a smaller logic tree containing those possibilities which are consistent with the current measurements. This is referred to as the diagnostic tree, which is a logic tree containing information about the current state of the actual transient.

To illustrate what an LFM model looks like, consider the simple system shown in Figure 1. This system consists of a pipe which normally carries fluid between two tanks at room temperature. Process disturbances can induce a change in the inlet fluid temperature. This change will shortly appear at the outlet given that the stop valve is at its normal open position. The flow can be stopped by closing the stop valve in the middle, in which case changes in the inlet temperature will not induce a corresponding change at the outlet.

The LFM diagram for this system is also shown in Figure 1. Each circle on the diagram is called a Continuous Variable Node (CVN) and represents a continuous parameter of the plant (in this case the inlet and outlet temperatures of the pipe). The square shape element is a Boolean Value Node (BVN) which represents the position of the valve. The value 0 is used for the normal (open) state of the valve and 1 for the abnormal (closed) state. The arrow shape element connecting the CVNs is a causality modeling element called Multiple Gain Box (MGB) which represents the causal relationship between the parameters of the system. The diamond shape element is a Test Box (TB). It represents the conditions which must be satisfied for the MGB box connected to the TB to be activated. Inside each MGB box is a gain value. Given that the MGB box is activated, the gain provides a linear mapping of the input to the output.

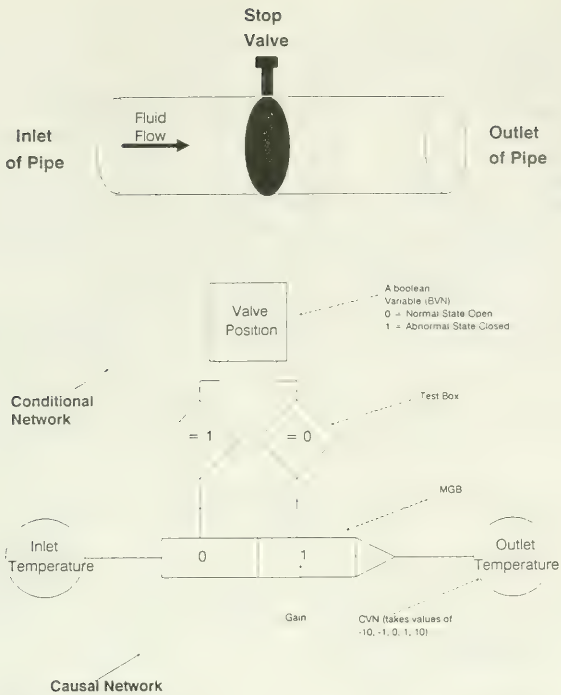


Figure 1. Example System and Associated LFM Diagram

The content of the LFM diagram can be expressed as follows. Given that the valve is open, a decrease at the inlet temperature will lead to a corresponding decrease at the outlet temperature. Given that the valve is closed, changes at the inlet will not lead to changes at the outlet temperature. The first statement is based on the MGB box on the right (the one with gain equal to 1), and the second statement is based on the MGB box on the left (the one with gain equal to 0).

TIME FLOWGRAPH METHODOLOGY (TFM)

A new approach, called the Time Flowgraph Methodology (TFM), has been developed to incorporate the time element to make the model more robust and to eliminate the inconsistencies that arise due to the presence of process loops in the time-independent approach. To accomplish these objectives, the LFM model has been "fuzzified" by treating the magnitude of the perturbations as fuzzy quantities. This automatically fuzzifies the notion of transitions between these states, and leads to an estimate of transition times (which are also fuzzy). These transition time estimates are used in conjunction with the underlying causal model to establish causal links between the observed perturbations. Furthermore, the use of the history of transition states has been introduced rather than a fixed set of

plant measurements. The diagnostic procedure can be used to identify the initiating events which can then be handed over to a symptom-based expert system for fault diagnosis.

The Time Flowgraph Methodology introduces the following extensions to LFM:

- o The time that it takes for a cause to induce its effect is taken into account. This is accomplished by associating a fuzzy time delay with each causal link. These time delays are used as a filtering mechanism when establishing cause and effect relationships between detected events.

- o Since concepts such as "normal," "small increase," etc., generally represent fuzzy concepts, fuzzy variables are introduced to represent the perturbation states. Using linguistic variables for state description provides a way of measuring the strength of the transitions between qualitative states, and thus introduces a degree of robustness into the model that will otherwise be absent when using crisp mappings.

Introduction of these changes make it possible to develop a new diagnostic approach which can be summarized as follows:

- o Qualitative transitions in parameters are detected and stored in a database. Each transition is characterized by a transition strength. The transition strength is related to the values of the state membership functions evaluated at the end points of the transition band.

- o The causal model is investigated to obtain all potential explanations for each transition in the database. Each potential explanation (referred to as a cause/context solution) involves a cause (which is a state transition involving another parameter at an earlier time) and a context (which is a set of constraints on the perturbation states of other parameters).

- o The cause/context solutions for each event are compared with the actual record of the transitions stored in the database to see if any of the cause/context solutions can be matched with those transitions. This establishes cause and effect links between the recorded transitions. Furthermore, a strength (referred to as cohesivity) is associated with each causal link. The determination of link cohesivities is based on the extent of the match between the candidate cause/context solution and the recorded transitions in the database.

- o Once all cause and effect relationships are established, the event strengths and link cohesivities are used to identify events which are primary (a direct result of a fault) and those which are secondary (due to the normal propagation of other events induced elsewhere in the plant). This establishes the fingerprint of the fault which is used to identify the root cause.

The transition database together with its causal links and associated parameters (event strengths, cohesivities, etc.) is called the Time Flowgraph.

As a conceptual example, consider Figure 2. This example represents a disturbance which was initiated by a sudden excess boron injection into the primary loop of a nuclear reactor. This initiated a reduction in neutron flux, which in turn induced a withdrawal of control rods and later returned the neutron flux to its normal value. The Time Flowgraph consists of a number of history lines representing the qualitative history of each of the plant's measured parameters. Each history line contains a record of all qualitative transitions that have occurred over the history of that parameter. The y-axis for each graph (though omitted) is implicitly assumed to correspond to the membership function for the time of those transitions. This means that each "bump" on a history line corresponds to a single transition and it represents the membership function for the time of that transition (i.e., each bump provides a fuzzy measure of the time at which the transition has occurred).

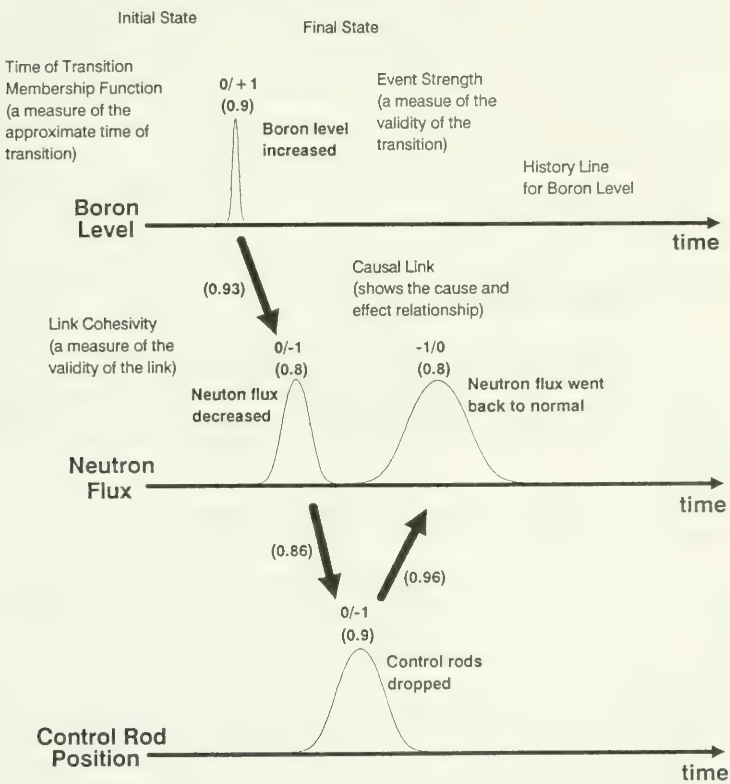


Figure 2. Reactor Disturbance Induced by Excess Boron

SOFTWARE MODULES

The methodology described in this paper is being implemented into EASE+ [4]. EASE+ consists of two parts: a) a high level software tool-kit for development of graphical interfaces to process monitoring or engineering analysis applications and b) a runtime software module that functions as a delivery environment for these applications. Using the interactive tool-kit, a developer can create full-color dynamically updated schematic diagrams, generate the necessary database structures, interface with external programs, implement the logic flow associated with a specific application, etc. With the EASE+ run-time module, an end-user can interface with an application through the graphics, menus, and data entry forms. EASE+ is being widely used in the process industry in general and the power industry in particular.

The TFM capability is being integrated with the expert system capabilities already available with the connection between EASE+ and NEXPERT [5]. This will result in the capability of performing a combination of heuristic and causal reasoning. Interactive graphics tools are being developed (using EASE+) for implementation of the TFM models. These tools will facilitate the development of TFM models to the point where a wide range of engineers will be able to develop these models.

An overview of the major software modules which perform the TFM functions are shown in Figure 3:

- The TransitionDetector analyzes the time-dependent plant data and makes a qualitative record of transitions in the state of plant parameters.

- The BackTracer, EventMatcher and LoopResolver establish cause and effect relationships between the transitions.

- The RootCauseIdentifier identifies those events which are primary (introduced directly as a result of the fault) as opposed to those which are secondary (introduced as a result of the causal propagation of other disturbances). In addition, it identifies the root cause based on the pattern of the primary disturbances.

Figure 4 shows the highlights of the control and data structure for the diagnostic module. The following procedure is used:

- Look at the instrumentation signals and record in Transition Database any transitions which have occurred. Continue until a user request (through an interrupt) is made to investigate the problem. This is accomplished by TransitionDetector.

- Investigate the cause of all events recorded in the database. This is done by calling BackTracer and EventMatcher. Back-Tracer looks at the causal model and provides all potential causal explanations for the event. These are then passed on to EventMatcher which compares them with the set of recorded transitions in order to establish causal links between the events.

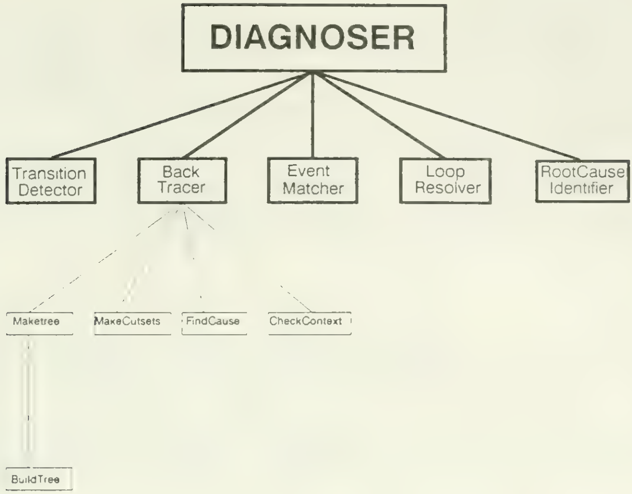


Figure 3. Structure of Diagnosis Module

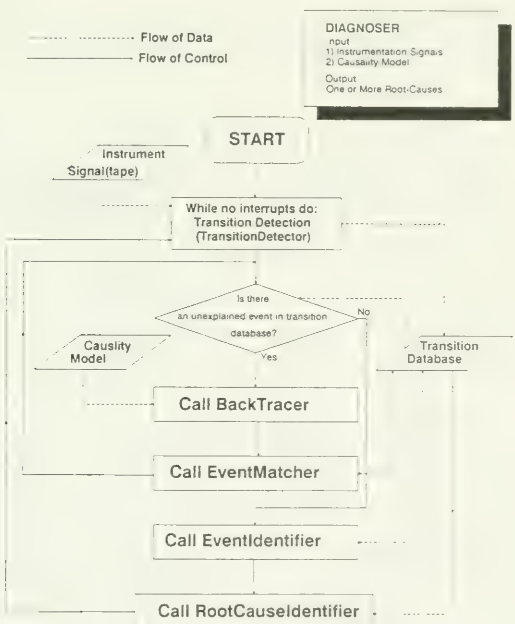


Figure 4. Control and Data Flow for Diagnosis Module

- Once all events have been processed, the EventIdentifier looks at the Transition Database and resolves any Time Flowgraph (causal) loops which might have occurred. It then identifies the extent (belief) to which each event can be considered a primary event. These beliefs are used by RootCauseIdentifier to establish the root causes of the disturbance; the results are then presented to the user.

- Once this is done, the Diagnoser returns to event detection mode; i.e., it starts analyzing raw data, and waits for another user interrupt to start diagnosis again.

EXAMPLE

Figure 5 shows an example of a disturbance scenario. The Time-Flowgraph in Figure 5 contains a causal representation of what has taken place. Included in the Time-Flowgraph are all of the detected transitions as well as the cause and effect relationships between these transitions. To avoid crowding the picture, event and link strengths are not shown; also different types of lines are used in order to make tracing of the picture easier.

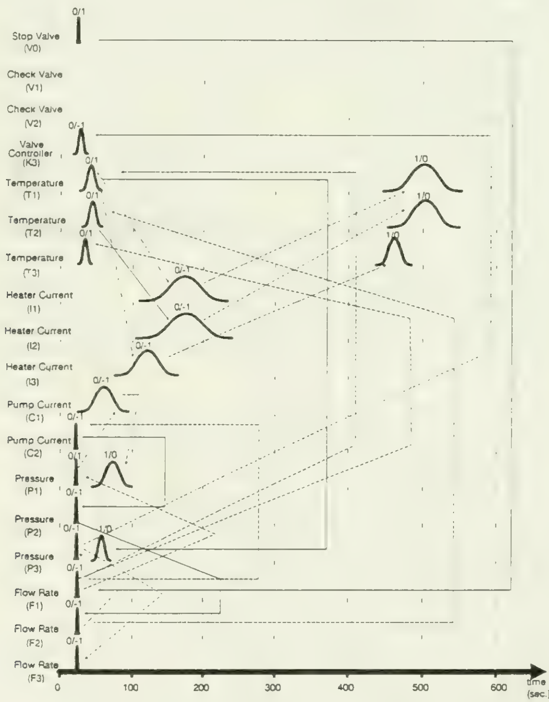


Figure 5. Sequence of Events following the Closure of a stop Valve

The only primary event (i.e., the event without a causal explanation) is the transition in V0, representing a closure of this valve. This has reduced the flow rate F1, which has induced a whole shower of additional effects in many other plant parameters.

Figure 5 is a representation of the analysis which takes place within the TFM software. As the software is completed, the complex aspects of the analysis will be made transparent to the user. However, he will interface with TFM through a graphical piping and instrumentation diagram (P&ID) representation of the plant and he will be able to request the software to perform diagnostic analysis and produce explanations of the results.

REFERENCES

1. R. L. Moore, M. A. Kramer. "Expert Systems in On-Line Process Control", Proceedings of Chemical Process Control III, Asilomar, CA, January 1986.

2. Sergio Guarro, David Okrent. "The Logic Flowgraph: A New Approach to Process Failure Modeling and Diagnosis for Disturbance Analysis Applications", Nuclear Technology, April 27, 1984.

3. Sergio Guarro, David Okrent. "The Logic Flowgraph: A New Approach to Process Failure Modeling and Diagnosis for Disturbance Analysis Applications", Final Report UCLA-ENG-8507.

4. EASE+ Technical Description, Version 3.0, Expert-EASE Systems, Inc., June 1989.

5. NEXPERT Technical Description, Produced by Neuron Data, Inc., 1989.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy as Small Business Innovative Research grant under contract DE-AC03-87ER80536.

Application of Artificial Intelligence for Nuclear Power Plant Surveillance and Diagnosis Problems

B. MONNIER, B. RICARD, J. L. DOUTRE, C. MARTIN-MATTEI, and A. FERNANDES
Electicite de France
Research and Development Group
6 Quai Watier
78400 Chatou, France

ABSTRACT

This paper presents three expert systems in the field of surveillance and diagnosis of nuclear power plants. Each application is described from the point of view of knowledge modelling. Then, a general knowledge model is proposed for a class of diagnosis problems.
At the end, the paper shows the future frame of the surveillance of the nuclear power plant main components at EDF in which the greatest part of those expert systems will run.

INTRODUCTION

For a few years, ELECTRICITE DE FRANCE has developed the use of Artificial Intelligence technics especially in the field of Surveillance and Diagnosis of nuclear power plants in order to extend the existing data processing chains towards automatic or computer aided interpretation and diagnosis.
The first part of this paper presents three expert systems as aids to diagnose defaults or failures of the main components of nuclear power plants:

- MIGRE: diagnosis of loose parts and interpretation of mechanical shocks in the primary circuit,
- DIVA: turbine-generator diagnosis,
- RGL expert system: troubleshooting of electronic equipments of the control rod drive mechanism.

We show the main characteristics of these applications and we particularly focus on the knowledge models which are made.
The second part is double. On one hand, we try to show that it is possible to design a unique knowledge model for a class of diagnosis applications. On the other hand, we briefly describe the future frame of the main components surveillance (primary circuit, turbine-generator, reactor coolant pumps, ...) in which the greatest part of those expert systems will run.

MIGRE: AN EXPERT SYSTEM AS AN AID FOR LOOSE PART DIAGNOSIS AND MECHANICAL SHOCKS INTERPRETATION

A loose part is often the consequence of structure damages: its detection is a proof of these structure damages. These damages can be the result of strong efforts or of stress corrosion. Sometimes it is an object which has nothing to do with the primary

circuit: tools or parts of temporary sensors forgotten after maintenance operations. Whatever its origin, a loose part can move within the primary circuit because of the flow, down to areas where it is trapped. These areas called trap areas are mainly the steam generator waterbox, the bottom of the vessel and the head of the vessel. The same phenomenon can exist in the secondary circuit, the trap area is then the bottom of steam generators (secondary side).
Once trapped, a loose part shakes around because of the coolant flow. It gives birth to mechanical shocks on the walls of the circuits which are detected by a specialized monitoring system on each plant. This monitoring records signals issued from accelerometers which are fixed outside the circuit walls in the trap areas. The accelerometers are sensitive to shock waves which spread in the metal.
When the monitoring system detects a high level of signal on one or several sensors, it sends an alarm. It is the first sign that there is an abnormal situation and then it is necessary to interpret the sensors signals in order to diagnose the phenomenon.

Nature of expertise

This interpretation is made by experts of the domain. It consists of the identification of the nature and the location of the phenomenon.
Two types of phenomena can be detected:

- phenomena due to the common functioning of the plant (valve opening, control rod driving mechanism, pump startup ...),
- abnormal phenomena (loose part, incore guide tube rattling, substructures impacting each other).

At the end, the diagnosis must precise:

- whether the detected phenomenon is abnormal or not,
- the cause of the phenomenon: loose parts, substructures impacting each other...
- the precise location of the phenomenon,
- and, if it is a loose part, its weight and the damage risks.

To build their diagnosis, the experts call in different kinds of knowledge and data:
- theoretical knowledge about physical phenomena like propagation modes, disturbance sources, the dynamic behavior of structures, signal processing and interpretation,
- knowledge issued from special experiments: shock simulation, sensitivity studies, multiple path propagation, standard background noise related to nominal plant functioning, ...
- qualitative knowledge issued from plant observation and failure analysis: identification of trap areas and of standard noises and frequencies, ...

The identification of the cause of the phenomenon requires a statistical analysis of several parameters of accelerometer signals. The main ones are: excitations periods, excited frequencies, time between impulses, delays between sensors and rise-time, duration and range of impulses on each sensors.
When the cause has been diagnosed, it is possible to determine the location of the phenomenon.

Knowledge model

In order to keep up with the requirements induced by expertise analysis, the knowledge modelling has split knowledge into two main parts: descriptive knowledge and reasoning knowledge.
The first one represents material or measurable entities (sensors, areas, signals, impulses, ...) and abstract entities like those which correspond to diagnostic concepts of the experts.
The second one represents the expert way of interpreting and diagnosing.

Descriptive knowledge. The expert universe is modeled by a set of entities precisely defined with their properties and their relations.
An example of entity (or object class) is SENSOR which is defined by its properties such as: NAME, LOCATION, VALIDITY, COINCIDENCE RATE, ...
Each property is an element of knowledge. It can be a simple value or a relation to another entity (ex: the location of a sensor is an area which is defined as an entity). The methods to obtain the value of a property are defined at its level: through inference, calculation, operator questions or access to a data base. We also associate some coherence controls on the value of a property: belonging to a list or an interval of values, control of the validity of this value if other ones are not known.
Among entities, we distinguish those which are called static and those which are called dynamic. The first ones represent all the objects which are permanent and not dependent on the phenomenon to diagnose (ex: sensor, area,...). The second ones depend on the phenomenon to diagnose (ex: impulse, signal, ...). More precisely, it is only the instances of dynamic entities which are created during the diagnosis session.

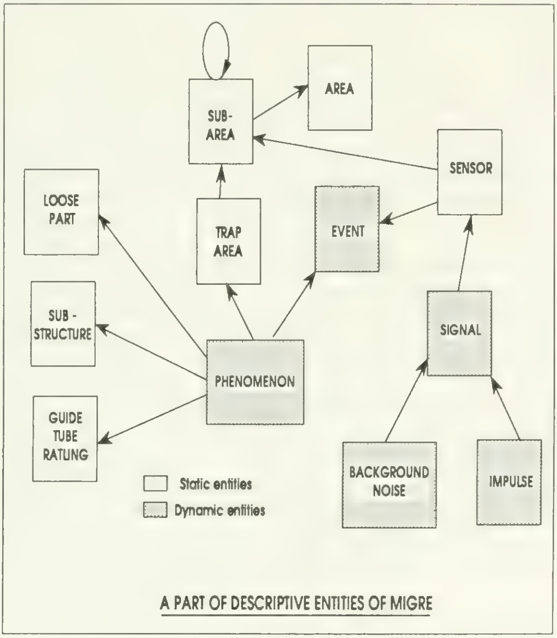


Figure 1: A part of MIGRE descriptive knowledge model

Reasoning knowledge. In MIGRE, the reasoning knowledge is less structured than in DIVA or the RGL expert system. The diagnosis process is modeled through production rules. Nevertheless there is a general guide which classifies the set of rules. They are functionally gathered by objectives: one group of rules is made for diagnosing failures of sensors, one for identifying the cause of the phenomenon, another one for the location and the last one is composed of meta-rules which drive the inference engine. The rules work on the entities described in the descriptive knowledge. In this way, it is possible to conclude independently on different properties of one entity: for example, it is not the same rules which diagnose the cause and the location of the phenomenon entity. At the end of a session, it only remains to read the value of the properties of the phenomenon entity to have the diagnosis.

An interesting particularity of MIGRE is that reasoning rules are written in natural language (in French). The French expression is interpreted in relation to the descriptive knowledge before being translated in an internal form.

Implementation and current state

MIGRE has been developed in PROLOG. The natural language interpreter and the inference engine have been specifically designed.
The expert system runs on micro-computer (Macintosh) and on a Unix workstation.
The knowledge base deals with phenomena which occur in the steam generators and the vessel. It is composed of 100 objects and approximately 300 rules.

DIVA: A TURBINE-GENERATOR DIAGNOSIS EXPERT SYSTEM

Most of the time, the main symptom of an incident on a turbine generator is an abnormal behavior of vibration measurements along the shaft.
Experts have to identify the causes of such vibratory abnormalities.The diagnosis made by experts cannot be limited to the sole elicitation of a fault from a list of symptoms. They try to come up with a satisfactory description of the encountered phenomenon. The consistency of this description can serve as a confirmation of a proposed diagnosis. And the other way round, a diagnosis could be reconsidered or estimated to be insufficient if the description cannot account for every observed symptom, or if some symptoms that are expected, given the proposed description, are absent.

The interest of an expert system for this application was justified by:
- the complexity of the problem, induced by the large number of possible failure causes and by the imprecise elements of the reasoning process;
- the limited number of domain experts;
- the importance and the high cost of some incidents on a turbine generator;
- the fact that theoretical knowledge (that could give way to an algorithmic solution) is of little use in expert diagnosis; this implies that the best direction for an automatic system seems to be to elaborate knowledge models that will be strongly inspired by the way experts perform a diagnosis.

The system prototype has been developed by Alsthom (French turbine generator manufacturer), Direction des Etudes et Recherches of Electricité de France (Research and Development Division of the French Electricity Board) and Laboratoires de Marcoussis (Compagnie Générale d'Electricité Research Center).

Nature of expertise

An analysis of expert knowledge and reasoning exhibited some key points for the construction of the expert system:

1) the great variety of processed informations (vibratory measurements along the shaft, plant operation parameters, technical characteristics of the turbine generator, maintenance operations, historical data on the group or on groups of the same type), and the great diversity in the useful aspect of these informations (value, classes of values, existence or absence of a property, correlations between informations...).
It should be noted that the information is usually not used for diagnosis as available in a control room or on monitoring systems. Therefore, data abstraction can be necessary to extract the important element from "raw" data.
Some knowledge elements will also have some kind of imprecision and uncertainty attached to them.

2) the fact that the reasoning process is based upon a recognition mechanism: the expert knows a set of possible situations and tries to see how a given case matches these typical situations. Of course -and that is probably the most significant difference between an expert and an automatic system-, the expert can "imagine" new situations when experience is not sufficient.

Reasoning on a hypothesized situation must deal with two points:
a) how well does this situation match reality?
b) can I give a more precise (or better adapted) description of the situation?

The key point in knowledge modelling has been to find out a way of describing such situations and of arranging these descriptions in a structure that will allow the implementation of an efficient recognition mechanism. The main point in automated diagnosis has been to design an efficient way of matching these pre-defined typical situations with reality.

Knowledge model

Prototypes and defaults. In order to keep up with the requirements induced by expertise analysis, the main concept used for knowledge representation is the notion of prototype: a prototype is the representation of a typical situation studied by an expert during a diagnosis; such representations can evolve from a very general frame *(an incident in steady operation conditions)* to the recognition of a precise kind of problem *(a rubbing problem, a blade loss…)*.

Part of the knowledge is included in the description of these prototypes and of attached information. Another part of knowledge lies in the relations between these prototypes. The basic relation between prototypes is the hierarchical link "…is a prototype which precises…". It follows the "natural" refinement diagnosis method. An excerpt of such a hierarchy is given in figure 2.

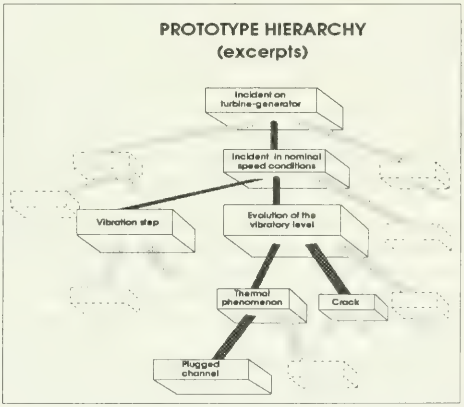


Figure 2: Excerpts from prototype hierarchy

A prototype can be seen as a schematic description of the mental representation of a typical situation by an expert. Therefore, it bears two complementary meanings:
- a prototype is a statement issued by the system on the current situation;
- a prototype is a unit where knowledge on how to reason further is stored: how to be sure that the prototype is relevant with a given situation, how it can be refined, what data should be acquired…

The final aim of the system is to identify the physical cause of the encountered anomaly.

Possible anomalies are called defaults. They represent the possible goals of Diva. Experts have identified thirty odd of them. Defaults are supposed to be evoked by prototypes describing their typical manifestations and to have the necessary knowledge to then arbitrate between the prototypes that have been recognized.

Descriptive knowledge. An important part of the knowledge used for diagnosis consists in knowing how descriptive information on the machine, its current situation and its past can be formalized. This includes technical knowledge on machines, phenomena physics, knowledge on plant operation and on maintenance, etc.
We must address:

- the encompassment of a large variety of data structures and relations,
- the need for a great flexibility in the way such data will be addressed (some requests might require a processing of relations between data).

This is treated through parameters which represent the basic elements used by Diva (e.g. the vibration that signals an incident), each parameter being described by attributes which hold its various characteristics (e.g. its frequency, its amplitude...). When necessary, parameter descriptions can be arranged in hierarchies, a given parameter descending along this hierarchy as diagnosis progresses and more details become available (e.g. a "signal" can later be seen as an "evolving signal", then as a "periodic signal").

Reasoning process. The global reasoning mechanism follows the following pattern:
Basically, DIVA considers a prototype at every stage in the diagnosis process, tries to check whether it is well adapted to the encountered situation, using already available data and, when necessary, complementary information which must then be acquired. A new and better adapted prototype will then be selected. It can be "better" by being the description of a more precisely identified situation if the previous prototype itself was "realistic" enough or by being a different situation, for instance if the previous prototype did not seem very well adapted or as an alternative possibility.
When various prototypes (corresponding to plausible descriptions of the situation) have been identified, the diagnosis module tries to synthesize these results leading to the identification of the most plausible defaults.
At every stage in the main part of its reasoning, DIVA first acquires a certain number of data which characterize this prototype; this is the data acquisition stage. Obtaining the useful information, a complex process including deductions, checks, transformations is sometimes necessary. Characteristic information on a prototype is described through components which put together a parameter, an attribute of this parameter, expected and rejection values for this attribute in the situation described by the prototype.
If there is no obvious reason for rejecting the prototype (i.e. if no component matches its rejection value), the system then seeks to confirm that this prototype is consistent with the available elements of problem description (or to reject the prototype as irrelevant); this is the confirmation stage. This confirmation can imply additional information acquisition (some might require special manipulations in the plant) in order to apply relevant rules.
The prototype will then be recognized with a certain confidence level (from rejection *"I am sure the situation described in this prototype did not occur"* to acceptance *"I am sure this situation did happen"*).
From the considered prototype, when it has been confirmed, the system tries to set up a new and more precise prototype using available data (either newly or previously acquired); for this new prototype, the process will be continued until a satisfying conclusion is reached. This is the prototype refinement stage.
If the considered prototype is rejected, the system tries alternative prototypes that have been left aside in the first place (e.g. siblings in the refinement hierarchy). The same research of alternative solutions is applied.
A complete diagnosis session of DIVA brings out a set of prototypes with various levels of confidence. As diagnosis knowledge and relevant information on the turbine generator

are associated to a prototype, this knowledge is useful to keep track of the reason why a prototype was rejected or accepted. Another element to interpret is the refinement chaining of prototypes that has been followed (e.g. how did the adequacy between prototypes and reality evolve when a more precise prototype was selected?). Figure 3 displays the main elements used in the model of DIVA.

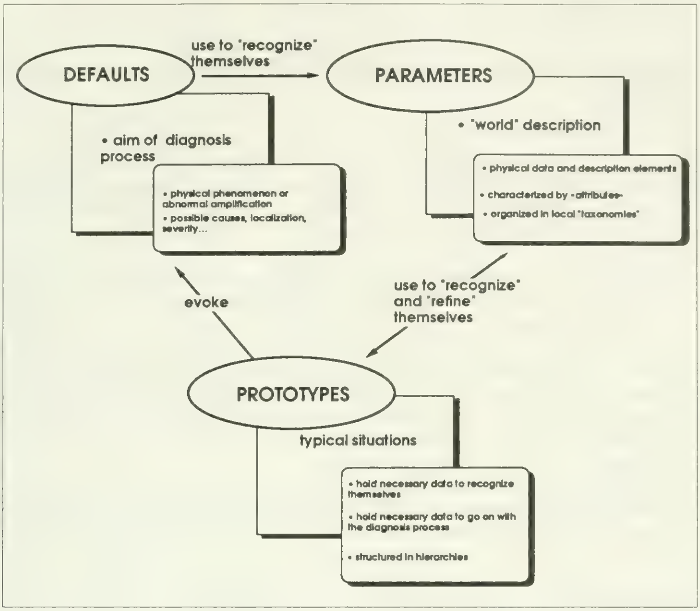


Figure 3: Main elements of DIVA

Knowledge "chunks". This designates basic knowledge elements which are directly related to the diagnosis process itself, as opposed to the descriptive knowledge which can have a wider use. This knowledge can be gathered into categories:

1) data abstraction knowledge, to obtain a piece of information useful for diagnosis, through processing of other data (such as basic information elements).
2) knowledge for prototype confirmation,
3) knowledge for prototype refinement,
4) knowledge for default confirmation.

These knowledge elements are usually expressed through local rule bases.

It is important to note that these mechanisms for refinement and for diagnosis and therefore the associated rules are only meaningful in a given prototype or default. This is why rules are associated to the task they contribute to (confirmation or refinement) and located in the prototype or default in which they are applicable.
Another point of interest in separating the way data is acquired from the reasoning of the expert system, which includes the way it will be used, is that both might evolve independently.

Implementation and current state

The prototype of Diva was first restricted to incidents in nominal speed conditions. Concurrently with the testing of these incidents, the system was extended to encompass

incidents in start-up and slow-down of turbine generators. It will soon include incidents in special circumstances (such as house load, turning, overspeed...) and incidents that only appear on long range observation (therefore bypassing speed conditions).
In its present state, Diva can deal with 32 defaults, includes 87 prototypes, 144 parameters described through 550 attributes. The various reasoning elements use some 2,000 rules.

RGL EXPERT SYSTEM FOR TROUBLESHOOTING OF ELECTRONIC EQUIPMENTS OF THE CONTROL ROD DRIVE MECHANISM

The control system of the control rod drive mechanism of a 900 MW PWR is a major electric-cum-electronic equipment in the 900 MW nuclear power plants. This complex logical element whose repair is difficult, is composed of control cabinets whose failures directly affect the availability of the plant.

Repairing this piece of equipment is still a problem for the operators as there is no well-defined diagnostic method, as failures are too rare to give them the knowledge necessary for their diagnosis.

The complexity of the equipment and its operating constraints satisfy the conditions for the development of an expert system as an aid for troubleshooting which provides effective diagnosis assistance and at the same time stabilises knowledge and the diagnostic process. The expert system helps maintenance technicians in their work in the event of a breakdown of this equipment.

The objective is to derive as accurate as possible a diagnosis and localization of the faulty components of the equipment from the fault signals produced by the equipment during a breakdown.

Nature of expertise

The diagnosis of fault signals involves applying reasoning mechanisms to the search for the cause(s) of equipment breakdowns. These mechanisms use two types of knowledge:

- knowledge of design which provides a functional structure of the equipment.This functional structure builds a distribution of the field of equipment into functions and sub-functions representing separate sets playing a specific role during operation,
- knowleldge of operation which describes the diagnostic process followed in order to identify the failures of equipment from available information.

Knowledge model

Both families of knowledge are described simultaneously in the knowledge representation model.
The model adopted is a diagnosis graph (figure 4) expressing the cause-to-effect relations in the sense of malfunctions which exist between the different equipment components involved in the signalling of faults.
The graph materialises, on one hand, the various equipment components involved when particular fault signals are triggered, and on the other hand, the relations between these components. More specifically, the elements of the functional structure appear as the nodes of the graph, and the relations between nodes as the propagation path of the malfunction. Diagnostic knowledge allows the path within the graph and the movement from one node to the next to be followed.
The point of entry into the graph is one (or more) fault signal(s) showing up on the signalling plates of the equipment. Displacement within the graph reflects the steps

of reasoning of maintenance technicians, and the localization of the element responsible for a breakdown is refined according to the results of tests performed by the operator on the equipment.

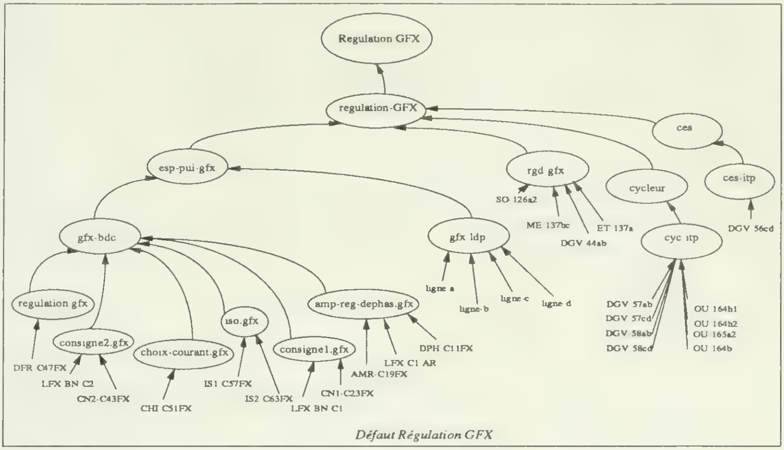


Figure 4: An example of diagnosis graph of a fault signal in the RGL expert system

All the graph components are elements of the functional structure of the equipment (function or terminal element). They serve as the medium for the description of the different reasoning steps during the diagnosis and correspond to an increasingly finer partition of the equipment.

There are priorities in the cause-to-effect relations between each node. These priorities, determined by diagnosis knowledge, define the order of exploration of the arcs of the graph. They can be defined a priori and dynamically updated during the diagnosis. So fields of research can be eliminated directly and dynamically depending on the knowledge base, the special conditions of repair and the information acquired.

The depth of the diagnosis provided by the expert system is determined by the localization of the smallest separable equipment component, e.g :

- a faulty module or group of logical electronic modules (and, or, etc...),
- a card which is defective or unusable,
- a faulty large electronic component (thyristor, diode, shunt, etc....).

Local diagnosis. So as to carry over from one graph node to the next in order to make headway in the search for the faulty element, a local knowledge base is associated with each node. The unfolding of the local diagnosis is in two steps with parameters depending on the context in which the fault signals appear:

- a "test" step allows conclusions to be drawn as to whether a node is operating correctly and therefore to identify the proper or defective functioning of an element of the functional structure. A test which does not find that a group is properly operating entails the suspicion of malfunction,
- an "orientation" step conditioned by the result of the first test determines the priorities driving exploration of graph paths according to the information acquired and the contexts relevant with the fault signals processed.

Additions to expertise. Additional knowledge is brought to bear, covering mainly:

- the use of shortcuts in the diagnostic graph: all the functional structure levels of the equipment are used in the selected model. There is therefore no direct

relation between the top element of a graph and one of its terminal elements. A more directional reasoning is used to study certain fault signals, especially fuse failures. This affects the representation of the model.
- test elements: in addition to graph components, expertise requires information relating to test elements which are not formally part of the functional structure and the diagnosis graph. These test elements can have several states and characteristics. The operator is only requested once and for all to supply information on them ; this is memorised. In a supplementary functional and material description of the equipment, the information is expressed by structured objects which represent the test elements.

Assumptions for the expression of knowledge. The expression of the diagnosis knowledge of the expert system is based on the following assumptions :
- failures are simple and stable,
- failures are multiple and stable providing they are formed of independent failures which can be diagnosed on the basis of simple failures in several cycles, using the system dynamics.

Implementation and current state

The complete expert system deals with 70 malfunctioning signals. It contains approximately 350 graph nodes, 1,750 terminal elements, 800 test elements and 2,500 rules. It is developed with the DIAGNEX tool based on the LE_LISP langage and using both an object oriented and a logical inference engine.The expert system will operate on an IBM PC extended with an 8 Mo card.
The average scanning time of a graph, excluding response and test time, is approximately one minute long.

METHODOLOGICAL EVALUATION

In spite of differences of objectives and domains, these three applications have great common points:
- they deal with diagnostic problems,
- they work off-line,
- they have an important part of interaction and dialogue with the user,
- knowledge is that of experts and specialists of the studied domains.

In addition, each of them points out the almost absolute necessity to design a knowledge model of the domain before beginning the realization itself of the expert system because the knowledge expression can only be efficient and coherent if and when this model is precisely built.
The objective of the knowledge modelling is to describe the domain and the resolution method of the problem. It allows:
- the description of the entities of the domain and their relations,
- the definition of the resolution method (ex: diagnostic process),
- the elimination of implicits,
- a control of coherence,
- the modularity and the robustness of the knowledge expression.

Designing a knowledge model gives another concrete advantage: this model becomes a kind of language common to experts and cognitive engineers. It is a means of communication. The more structured and precisely defined the knowledge model is, the more efficient and coherent the communication and the knowledge expression are .
Two types of very different situations can occur when designing a knowledge model:
a) this model is directly based on a functional or material description of the equipment because the studied knowledge domain is strongly linked with these descriptions,

b) the knowledge domain has no link with the functioning or the structure of the equipment and it is necessary to design a specific model adapted to the problem.

Whether you are in one or the other case is far from being alike when realizing an expert system.
The b) situation occurs when there is a gap between the knowledge domain which is studied (ex: vibrations) and the nominal functioning or structure of the equipment (ex: turbine generator or steam generator). In this case, the experts of the knowledge domain are not those who designed the equipment itself; knowledge is difficult to reach (lack of written documents, implicit know how, ...). Then the knowledge modelling is critical, often long and uncertain: the knowledge expression can only be efficient and coherent if and when this model is precisely built.
At first sight, the a) situation is easier because it is based on explicit elements (written documentation, functioning models, material structures, ...). It should lead to a better efficiency for developing expert system applications.
The three applications previously described belong to b) situation.

Another parameter has a great effect on knowledge modelling: that is the automatic access to data or not which are necessary to the expert system. When the expert system automatically gets data, the resolution method can be simplified even if the computer has to work more: it is possible to deduce all the possible conclusions by means of a systematic and exhaustive use of data and knowledge. On the opposite situation, data are entered by an operator and the resolution method has to define the strategy of data acquisition through questions which must be asked at the proper time and with a proper schedule and be understood by the user. Then the resolution method can be more complex (assumption expression, backtracking, non-monotony, ...) in order to decrease the number of questions or to take the difficulty to obtain data into account for example.

Designing a knowledge model is a critical stage during the development of an expert system. Its duration depends greatly on the complexity of the knowledge domain. But to be able to establish this model is the first sign that it is possible to realize the expert system and can be an indicator of the volume and the complexity of knowledge. During this stage, the effort is focused on the analysis of the knowledge domain without reference to the tools or languages used later to implement the expert system. The knowledge model allows to choose or to define the development tools or the knowledge representation formalisms.

TOWARDS A UNIQUE KNOWLEDGE MODEL FOR DIAGNOSIS EXPERT SYSTEMS ?

Because of the importance of the knowledge modelling, it would be very efficient to rely on general models for each class of problems to realize expert system applications. The three above applications belong to the same class of problems which can be called "Technical Diagnosis by Observations". They suppose that causes and effects remain stable during the diagnosis and for which the diagnosis method is made by means of observations and does NOT imply important structural or functional modifications (ex: complete stop of installation, heavy disassembly, destructive inspection, ...).
Moreover, we think that from those three applications, it is possible to point out a unique knowledge model.
All three applications, as a matter of fact, imply using three levels of knowledge:
- a descriptive level,
- an inference level,
- a strategic level for reasoning (diagnosis).

Descriptive level.

The first need when modelling knowledge is to make an exhaustive list and define the

entities which compose the domain and also the problem which is to be solved.
Those entities are either concrete when they correspond to material or measurable elements (ex: sensors, parts of equipment, physical variables, ...) or abstract when they represent concepts peculiar to experts or to the diagnosis method (ex: reasoning steps, prototypes, phenomena, ...). Modelling such entities is done with a concern for structuration, accuracy and gathering in generic classes. The use of object oriented technics (especialy frame based) is now standing for this level of knowledge.
From a methodological point of view, the most important is to define the "slots" of the structured objects properly without conflicts nor ambiguity with the other levels of knowledge. An object is an autonomous whole in that it has in itself every thing necessary to its manipulation by the other levels of knowledge. For example, the method to obtain the value of the slots of objects must be defined: through inference, calculation, access to a data-base, questions to operators, ... If this is done through questions, there is left to decide whether the text of the question is defined at the level of the class of the object or at the level of the object itself.
The descriptive knowledge level also deals with relations between entities. Those relations depend, of course, on applications; the most usual ones are:

- hierarchical relations such as sub-classes to classes,
- belonging relations such as "is-a",
- simple relations such as "refer-a" (ex: an object slot has not a value directly but refers to another object),
- relations of composition such as "part-of".

The aim of the descriptive knowledge level is easy to understand and "natural", but carrying it into effect is more delicate and requires a lot of care and method.

Inference level and strategic level for reasoning.

The strategic level of knowledge modelling corresponds to the design of the method to solve the studied problem. It therefore directly depends on the aim of the application. The fundamental point in the description of those strategic and inference levels is that knowledge elements used here serve a precise objective (within the strategy) and should be associated with this objective.
For diagnosis problems, our experience suggests that the diagnosis process might be based upon a top-down "classification" approach. By that, we mean collecting relevant information (observations, external symptoms,…) to proceed through successive steps towards identification of correct diagnoses. Concretely, such knowledge modelling can be represented through a classification graph (or a set of graphs with a specific contribution to the overall diagnosis process each). Roughly, each node of the graph corresponds to a reasoning step. The relation between two nodes is a relation of refinement: the more you advance, the more precise you become. Depending on the application, you must rely on specific analysis methods to build this graph and express the nature and meaning of moving along its paths.
In the RGL expert system, the graph nodes represent the functions of the equipment and the type of their relations is a causality relation: the failure of the slightest function involves the failure of the most general function. The graph is used backwards of the causality relation for the diagnosis.
Each node should be provided with the set of local partial goals that are to be treated when examining the step represented by the node (confirm, invalidate the reasonning step represented by the node, decide of orientation), the relevant knowledge aimed at attaining these goals (usually rules) and the inference strategy that should be applied to these rules in order to reach these goals. These local knowledge bases compose the inference level. In addition, the items addressed by these knowledge bases are the objects and properties described at the descriptive level.
The strategy graph defines the diagnosis process as a whole and the local knowledge bases and associated inference strategies describe the graphe course.
Such strategy graphs can be implemented with object-oriented programming techniques, allowing a better modularity, flexibility and readability of the implemented

structures. If the reasonning is homogeneous along the whole strategic pattern, the local inference process might be "factorized". Knowledge bases are usually implemented as sets of production rules associated with an adapted logical inference engine. The whole knowledge model can be roughly represented by figure 5.

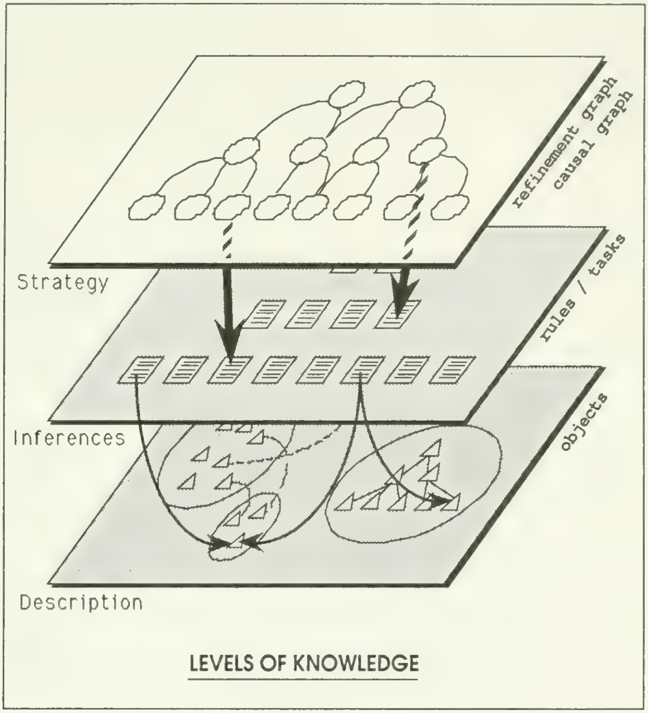


Figure 5: Knowledge levels

A few precisions can be given to write those various forms of knowledge:

- each level of knowledge must be autonomous and must not interfere with the others,
- the way to obtain the value of the object slots must not be defined by the rules bases,
- the rules bases must not modify the strategy graph.

Such a knowledge model answers the aims of knowledge modelling given in the knowledge evaluation chapter and it is a flexible guide to the designing of expert system applications belonging to the "Technical Diagnosis by Observations" class.

THE FUTURE FRAME FOR SURVEILLANCE AND DIAGNOSIS EXPERT SYSTEMS AT EDF

Together with the development of Surveillance-Diagnosis Expert Systems (which focus on knowledge representation), ELECTRICITE DE FRANCE is currently designing a new surveillance and diagnostic concept which will combine many state-of-the-art diagnostic technologies in surveillance and condition monitoring.
This new concept leads to the design and the development of an Integrated Diagnostic

Aid and On-Line Monitoring Center (PSAD in French is the acronym for Poste de Surveillance et d'Aide au Diagnostic). The PSAD in the first stage of the study will be devoted to the on-line monitoring and diagnostic of the main components of nuclear power plants (Nuclear Steam Supply System, Steam Turbine, Main Reactor Coolant Pumps).

The PSAD will call in the state-of-the art in terms of data processing, advanced computers, data management and artificial intelligence. In the PSAD, the place of AI applications will be mainly at the end of data processing chains in order to extend these chains towards abnormal situation interpretation and malfunctioning diagnosis. As the PSAD will manage surveillance data, it will be possible to link these surveillance data and expert systems (which is not the case to day). So it will be possible to reach automatic interpretation and diagnostic and not only computer aided interpretation and diagnosis.

Objective of the PSAD

The main objective of the PSAD is to provide plant personnel and experts with an efficient and friendly-to-use decision-aid for the detection of failures occurring on the main components of nuclear power plants.
Primarily the PSAD is a tool for operating teams.They will get services such as: monitoring data processing, diagnostic modules organized by monitoring functions (e.g: turbine monitoring, primary circuit monitoring, reactor coolant pumps monitoring,....) which will enable them to cope with abnormal situations or malfunction of monitored components.
Then, the PSAD is devoted to specialists and experts who have to deal with difficult or not yet identified cases and who will find in the PSAD, monitoring data and data processing which are necessary either for their remote analysis or for their work at the plant location.
Artificial Intelligence (Experts Systems)will be a part of the diagnostic modules available on the PSAD in order to provide an assistance to operating teams for diagnosis and data interpretation.

The PSAD is designed with a flexible architecture in order to handle variable monitoring functions.
For the first version, the following monitoring functions and systems will be covered:
- Loose parts detection and localization
- Turbine-generator vibrations
- Reactor coolant pumps vibrations

These functions were selected, first because the monitored equipments are safety or availability related, then because the data acquisition systems associated to the monitoring equipments provide a very large amount of data difficult to interpret directly.

According to these analyses, the design of the PSAD put an emphasis on the following items:
- Functionally,the PSAD will integrate the data and the data processing already existing on the previous systems and on the plant computer.
- It will provide new capabilities in terms of data processing needed for surveillance and diagnosis in order to obtain homogeneous softwares.
- All the management of the various tasks will be automated as much as possible.This will be particularly true for data acquisition and for their management. All the data provided by the monitoring systems will be automatically collected and stored without any human intervention. In addition, the status of the normal behavior of the hardware of the monitoring systems will be permanently checked.
  The on-line monitoring systems will display messages to the operators only in case of an abnormal behavior based on routine data processing. Additional data processing will be triggered in case of abnormal events.

- The monitoring systems will produce only digitized data.This will lead to the developments of entirely new systems.
- The PSAD will be connected to the EDF nation-wide data link.Thus if necessary, it will be possible to transfer rapidly raw and processed data to the experts of the Generation Division and of the Research Division

The PSAD architecture

The Integrated Diagnostic Aid and On-line Monitoring System has a structure which is organized with four levels (see figure 6):

- -the on-line monitoring systems
- -A master workstation
- -An analysis workstation
- -A remote workstation

The monitoring systems. They are close to the monitored components and they provide acquisition and real time processing of physical measurements. At their level, they make reduction and compression of the raw signals before their transmission to the master workstation.There is no interface available to plant personnel with the monitoring systems. So there is no monitoring nor diagnostic activity performed by operators on the monitoring systems. Only routine maintenance tasks are scheduled. Three up-to-date technologies monitoring systems are currently under design and specifications:
The SACP system (Surveillance Automatisée du Circuit Primaire) is dedicated to the monitoring of the Nuclear Steam Supply System.It includes the SMART data processor for loose part detection.
Two SAMT system (Surveillance Automatisée des Machines Tournantes) will monitor the turbine generator and the primary pumps.

The master workstation. It is the first and main level where operating teams work. All the automatic and real time data processing are implemented in it: monitoring alarm display, automatic pre-diagnostic, on-line computations. Operating teams are continuously informed of the status of the monitored components.
The operator will also have an access to the data processing they are interested in (mainly data displays).
The master workstation manages the acquisition and the storage of the data produced by the monitoring system using data bases updated in real time. All the maintenance and management operations (remote loading, configuration,....) of the monitoring systems are initiated from the master workstation.
There is one master workstation running permanently for each power plant unit.

The analysis workstation. It is devoted to off-line analysis of events, abnormal situations or past events.
This is a tool for monitoring and maintenance engineers. All the data processing modules are triggered upon request by the operators. Thus they have access to data stored in the data bases of the master workstation in order to get aided decisions.These decision aids are based either on algorithmic or expert systems techniques. The DIVA and MIGRE expert systems will run on this analysis workstation.
On a given plant site, there may be only one analysis workstation for several units. The communication between the analysis workstation and the master workstation is made by using a plant site local network.

The remote workstations. They are located at the headquarters of the Generation and of the Research and Development Division. Their designs are similar to that of the analysis workstation. They will be used by national experts or specialists of monitored components. These experts processed data which are stored in the master workstation and which are sent to the remote workstation when abnormal situations or incidents difficult to interpret have occurred. It will be possible to run the DIVA and MIGRE

expert systems on this remote workstation. Data transfers between the master workstation and the remote workstation are performed by the use of the EDF private national data highway.

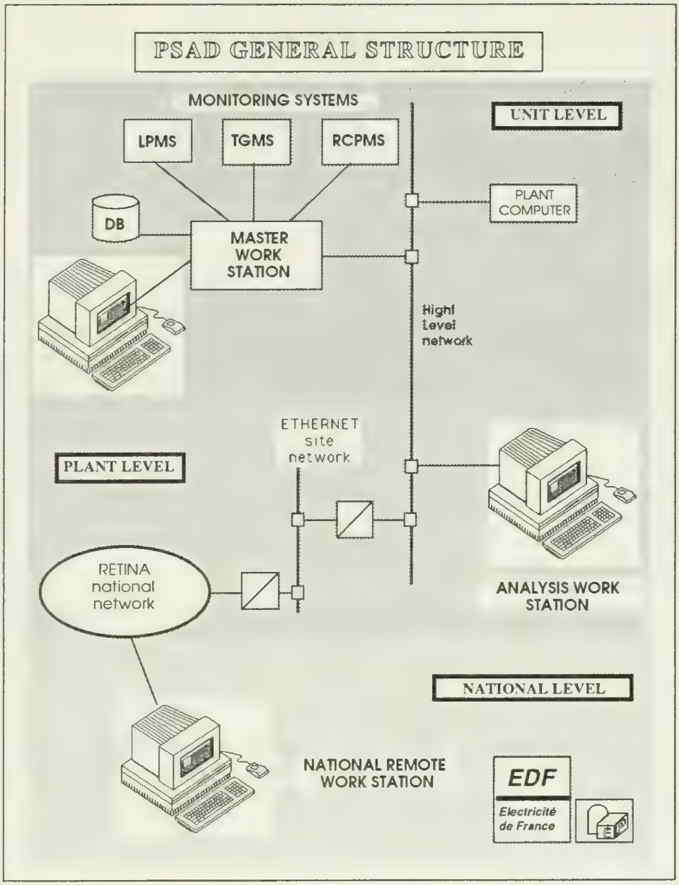


Figure 6: The PSAD structure

CONCLUSION

In order to improve the availability and the safety of its nuclear power plants, ELECTRICITE de FRANCE is developing both expert systems and an Integrated Diagnostic-Aid and One-Line Monitoring Center. The first ones demand great methodological efforts to model the necessary knowledge. The second one will allow to reach automatic interpretation and diagnosis because of the link between expert systems and surveillance data.
The PSAD is a significant effort made by ELECTRICITE DE FRANCE to increase the efficiency and the costs reduction of surveillance.

REFERENCES

Brunet, E., Fernandes, A., Martin, C. (1987) MIGRE: Système expert d'aide à l'interprétation de chocs mécaniques en centrale - L'Expertise dans MIGRE. Note EDF HP-A2/87-08

Aikins, J. (1983). Prototypical Knowledge for Expert Systems. Artificial Intelligence, 20 (2), pp.163-210.

Chandrasekaran, B. (1987). Toward a functional architecture for intelligence based on generic information processing tasks. 10th IJCAI, Milano (Italy).

David, J.M., Krivine, J.P., Tiarri, J.P., and Ricard, B. (1988). DIVA: Recognition of Typical Situations for Turbine Generator Diagnosis. Journal of Intelligent and Robotic Systems, vol.1, n°3. Kluwer Academic Publishers.

David, J.M., Krivine, J.P., Tiarri, J.P., and Ricard, B. (1989). DIVA: Système expert pour la surveillance vibratoire des groupes turbo-alternateurs. 1st Artificial intelligence convention (Convention IA) , Paris (France)

Kirk, D.B., and Murray, A.E. (1988). Verification and Validation of Expert Systems for Nuclear Power Plant Applications. EPRI NP-5978, Project 2582, Final Report

Osborne, R. (1988) On-line Artificial Intelligence-based Turbine Generator Diagnostics. AI Magazine, Fall 1986 issue, pp.97-103.

Ricard, B., Monnier, B., and Morel, J. (1986). Système Expert d'Aide au Diagnostic des Défauts d'un groupe turbo-alternateur. 2nd International Conference on Artificial Intelligence (CIIAM 86) , Marseille (France).

Ricard, B., Monnier, B., Morel, J., David, J.M., Krivine, J.P., and Tiarri, J.P. (1988). Overview of DIVA: an Expert System for Turbine Generator Diagnosis. In Majumdar, M.C., Majumdar, D., and Sackett, J.I. (Ed.), Artificial Intelligence and other innovative computer applications in the nuclear industry (American Nuclear Society Topical Meeting,1987, Snowbird, Utah), Plenum Press.

Monnier, B., Doutre, J.L., Martin, C., Système expert RGL: Détermination d'un modèle des connaissances. Note EDF HP-A2/87-06

Computerized Procedures—The COPMA System and Its Proposed Validation Program

METTE KROGSAETER, JAN SVERRE LARSEN, SVEIN NILSEN, WILLIAM R. NELSON, and FRIDTJOV OWRE
OECD Halden Reactor Project
Os Alle 11, 1750 Halden, Norway

ABSTRACT

The written operation and emergency procedure is an important information source for the control room operators of any complex process, including nuclear power plants. The OECD Halden Reactor Project is currently investigating how to computerise such procedures and how to design a good man-machine interface based on sound ergonomic standards for an on-line, computerised procedure system. The aim is to clarify whether such a system can improve safety.

A computerised procedure system, called COPMA (COmputerised Procedure MAnual) has been developed and integrated in the Halden Project's advanced experimental control room facility HAMMLAB. The system is designed to replace hadrcopy procedures with a computerised manual. COPMA assists reactor operators by making the procedures available rapidly, displaying the needed procedure steps, performing on-line checks of plant parameters when required to make a decision. Other features of COPMA are: use of AI techniques such as the symbolic languages PROLOG and LISP, and use of LISP machines such as TI Explorer II. It is linked to a PWR nuclear simulator and other computerised operator support systems in the laboratory. An extensive human factor validation experiment is now under preparation.

The COPMA system is also installed at the Italian organisation ENEA's laboratories outside Rome where it is linked to a PWR nuclear power plant engineering simulator. A joint research programme beween the Halden Reactor Project

and ENEA is currently under way where emphasis is placed on structural analysis of operating procedures of different types of plants and human factors issues in connection with compuerised procedures.

1. INTRODUCTION

The structure of traditional written operating procedures is normally step and instruction based, i.e. the procedures consist of a set of steps, each step consisting of one or more instructions to the operator about checks to do or actions to take.

The scope of the computerised procedure manual (COPMA) is limited to a system that can handle today's written procedures in the context of both a conventional, but also an advanced computerised and CRT-based control room. The aim of the COPMA system is to assist control room operations in selecting a relevant procedure and to execute that procedure once it has been retrieved. The belief is that providing rapid access to stored procedural information and relieving operators of trivial tasks in connection with executing procedures such as collecting data, waiting for responses and doing responce checks, are important additional advatages.

The procedures available in COPMA can be of several different types, e.g. emergency procedures, disturbance procedures or normal operating procedures. However, the fact that the procedures are executable also implies that they are represented according to a specific format.

Procedures are entered in COPMA using the procedure editor PED, in which each procedure is given a textual and a graphical representation. The graphical replresentation is mainly used for displaying the procedure flow-chart, whereas the textual representation (a PROLA program) forms the basis of the instructions as they are displayed to the operator.

COPMA is a computerised and CRT-based system that enables operators to retrieve procedures from a procedure database and follow one, or several, procedures in parallel. It is used in the following way: the operator starts the execution by selecting the procedure he wants to execute from a list of all available procedures. A graphical overview of the procedure is then displayed to him as a tree-structure. The execution is from then on an interactive session, starting with the operator selecting a procedure step for execution. The content of the step is then displayued to the opereator as an organised, well structured list of instructions. The task of the operator is then to carry out these instructions.

1.1 COPMA Functions

The COPMA system is designed to replace hard-copy procedures with a computerised procedures manual. COPMA assists reactor operators by making the procedures available rapidly, displaying the needed procedure steps, and performing on-line check of plant parameters when required to make a decision. In addition, COPMA functions to keep track of the user's place within the procedures structure. Thus, COPMA displays a "map" of the operator's current position within th procedure structure. In addition, when the operator must branch to another portion of the procedure or access another procedure, COPMA will keep track of the position wthin the structure and remind the opereator the location to which he should return when the branching instructions have been completed.

1.2 COPMA Objective

The primary objective of COPMA is to reduce the risk of plant operations by:

1. Reducing the time required to access and implement an operating procedure.

2. Reducing the number of errors committed in procedure access and execution.

3 Reduce the likelihood of "getting lost" in the procedure system and thereby performing incorrect actions.

2. PROCEDURE LANGUAGE

For a computerised procedure system to function effectively, the procedures must be stored in the computer in such a way that they can be accessed and executed fast. However, such a structure is most probably difficult to comprehend for humans. It is therefore necessary to have a method whereby one can formulate the information in the written procedures in such a way that it is easy to read and understand by procedure designers/writers, and later is easy to translate into a form suited for execution by the computer system.

The language has a well defined syntax and semantics. Therefore it is possible to build an automatic translater from the PROLA input format to the internal format used by the computer system. This translater is called the PROLA compiler.

2.1 Preparing Executable Procedures for COPMA

The preparation procedure consists of four major steps: 1) reconstruction and rewriting of the procedure into PROLA format, 2) editing the procedure text and the overview graph by means of the procedure editor PED, 3) checking and translating of the procedure by means of the PROLA compiler and 4) construction of a cross-reference table called the Variable-Address-Table (VAT).

Step 1 is the key to a high quality computerised procedure. It is here that the foundation is made for a good layout and correct content of the procedure to be used on-line in the control room. This work should be carried out by responsible design and/or operating staff. If this step is not done properly, there is nothing in the other steps that can improve the quality.

The remaining three steps are in reality plain data entry/editing type of work by means of the COPMA OFFLINE program system. The work can be done by any type of staff since the program system is designed to be straight forward and easy to use.

2.2 Reformulating Existing Procedures

The process of transforming written procedures to PROLA format, is meant to by easy to understand and accomplish for personnel which is involved in process operation and procedure construction.

The basic idea is to write the procedure in a language which is built up around procedure steps as basic building blocks. Any number of instructions can reside inside each step, see Figure 1. These instructions are, of course, the actions and checks that the procedure designer wants the operator or computer to do at the actual moment, such as give/read some specific information, ask the operator to manually perform some checks, let the computer automatically check and evaluate some condition, let the computer monitor a process condition, branch to another procedure step etc.

The language consists of several reserved words. The basic PROLA-words are used for grouping of instructions into steps and for specifying the types of instructions to be executed. The words normally have some additional information connected to them which more precisely describes what should be done for each particular instruction.

```
                    P R O L A

Procedure <procedure-identifier>

         Step <step-identifier>
                  Instruction
                  Instruction
                  ..
         Endstep

         Step <step-identifier>
                  Instruction
                  ..
         Endstep

         ...
Endprocedure <procedure-identifier>
```

Figure 1. Procedure, Step and Instruction structure.

3. PED - PROCEDURE EDITOR

The COPMA off-line system is called the Procedure EDidtor (PED). This editor is a tool for entering plant procedures in the PROLA language, editing the procedure graphs and entering procedure related display formats and component manipulations into the COPMA system. The editor is based on a graphical window and a menu window. The user has to select commands from this menu to build procedure graphs or performing some other operations.

4. ACTIVITIES

Applying a procedure on the plant need not be a strictly sequential undertaking. It might well be that the procedure asks the user to do things in parallel. The reasons for this are quite obvious. A plant does not consist of isolated components, but all things depend on each other in order to function. For instance, if one wishes to do something on component A, it might also be necessary to perform some other action on component B. Sometimes, it is possible to descibe the working procedure as a pattern of interleaved actions on the compenents A and B. But not always. Occasionally the sequence depends on factors which are not easily described, or even easily defined.

Consequently, when the user applies a procedure which contains several parallel branches, it would presumably be

beneficial for him to have a system keeping track of how far he has come in each and one of the branches. In order to deal with this porob lem, COPMA is based upon the activity concept. An activity is not the same as a procedure, but it is subordinate to a procedure. A procedure might have one or more activities associated. Whenever the execution of a procedure splits into two parallel branches, a procedure gets two activities associated, one for each of the branches. As mentioned above, the active procedures pane will contain both the procedures and the associated activities. The user may then use the mouse device in order to pick out the activity associated with the branch he wishes to work with.

The activities themselves have some important attributes which make them very useful, not only in the case of parallel branches. To begin with, they have a history associated. This means that for each activity, it is recorded exactly which instructions and steps that have been performed. It is possible to inspect his history to see what has been done so far related to the activities.

Furthermore, activities may have one or several bookmarks associated. A bookmark is a pointer to a step associated with the activity. By performing a select operation on the bookmark, that step will be displayed both in the overview pane and in the step survey pane.

Another important attribute is the current instruction. This attribute is what makes the activity something more than a device for implementing parallel branches. The current instruction is a pointer which indicates how far the procedure has been taken. Put another way, the current instruction is the next instruction to be performed.

Procedures do not have any current instruction attribute nor any history. Of course, the content of procedures can be inspected, but if one wishes take advantage of the follow-up facilites offered by the COPMA system, it is compulsory to make activities.

The normal way of making activities, is by identifying a starting step in the overview pane. As soon as this is done an activity is created with the current instruction set to the first instruction in that step and with an empty history. Later on, both the instruction and history attribute is updated according to what the user chooses to do related to that activity.

Activities may also be created because they are specified in the procedures. One typical example of this, is when a procedure splits into two parallel branches. The execution of a socalled INITIATE instruction in the procedure may enforce the creation of an activity starting at the step

indicated in the INITIATE instruction. Whenever the user allows the execution of such an instruction, a new activity will be created. This activity constitutes a branch which is parallel to the branch already beeing executed.

Another specific type of activity must also be mentioned, namely the monitor-activity. These activities always originates from the socalled monitor instruction. A monitor instruction is an instruction which tells the system to monitor some plant condition continuously over time. Depending on the outcome of this monitoring an activity may be created by the system automatically. Such an activity may prescribe some remedial actions for the phenomenon detected during the monitoring. But, of course, it is up to the user to select this activity immediately, or postpone the execution of it because he is currently working on another more urgent matter.

Generally, the user will have to work with a whole set of activities. However, only one activity can be attended to at a time. It is up to the user to decide which activity is the most urgent one, and he will probably have to revise his opinion on this from time to another as things develop and new activities are created.

5. COPMA MAN MACHINE INTERFACE

In this chapter, some crucial aspects of on-line COPMA will be presented. The concepts introduced here are important for the understanding of the system.

5.1 Man Machine Interface

In COPMA, all input from the user is given with a mouse device. Feedback from COPMA is presented in various ways, depending on the user action.

5.1.1 The mouse

The mouse device is connected to the terminal. On the screen, there is a mouse symbol shown as an arrow. When the mouse device is moved around on a special surface for reading the movements of the mouse device optically, the position of the mouse symbol on the screen will change correspondingly.

When the mouse symbol is placed on a command field, the field will be highlighted, i.e. a rectangular box wili be displayed around the command. When the mouse symbol is moved off a command field, the highlighting will disappear.

That a field is highlighted means that a command can be executed.

The mouse device has three push-buttons on top of it, which are used for giving commands. A command is given by clicking the mouse, i.e. pressing a mouse button when the mouse symbol is over a highlighted item.

5.1.2 Scrolling and mouse documentation in panes

The COPMA screen consists of a set of non-overlapping windows, the most important ones will be called panes. The content in a pane may be larger than the available space on the screen. To take care of this problem, all panes but one have associated scroll bars, narrow windows placed next to the panes.

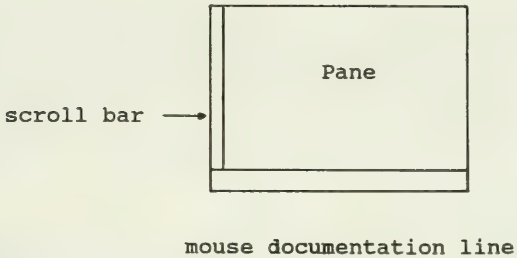


Figure 2. A typical COPMA pane.

Figure 2 shows a sketch of a pane with associated scroll bar and mouse documentation line. The mouse documentation line is used for informing the user about the actions that may be initiated by clicking the mouse. Consequently, each time the mouse symbol is pointing to a highlighted item in the pane, the mouse documentation line will inform about the significance of differnt mouse clicks.

5.1.3 Other aspects of the COPMA man machine interface

Other aspects of the man machine interface can be summarised as follows:

1) Feedback on mouse clicks.
2) Confirmation of mouse click.
3) Pop-up windows.
4) Audio signals.

5.2 Panes

A pane is a major window in the COPMA screen. The cointents of a pane will be updated dynamically as COPMA on-line is executed. Each pane has the responsibility for a set of operations necessary for the execution of procedures.

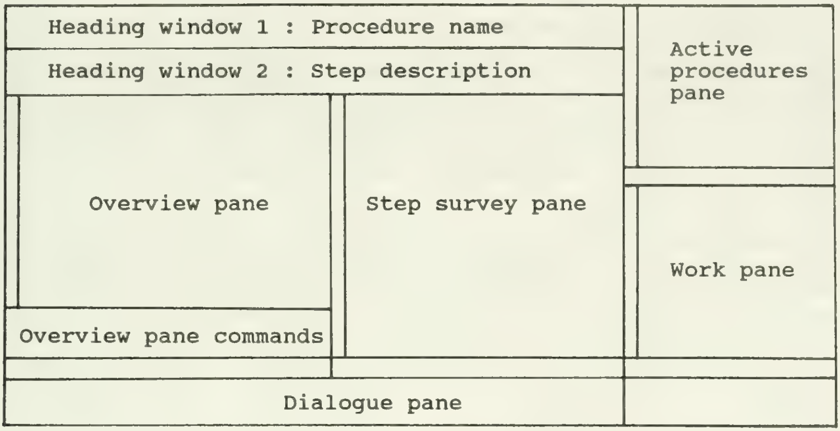


Figure 3. COPMA panes

Figure 3 shows how the COPMA panes and other windows are placed on the screen. In this section, we will present the features and the purpose of each pane.

5.2.1 Dialogue pane

There are three main purposes of the dialogue pane:

- offering some important commands
- informing the user about the status of the system
- informing the user about date and time

5.2.2 Work pane

The work pane will initially be used as a bookshelf where the procedure manuals can be found. However, the work pane will seve various purposes during a COPMA run session:

- presenting a list of all available procedures (manuals bookshelf)
- presenting a list of all active monitor elements
- presenting the values of process variables in auto check instructions
- offering an editor for entering and displaying comments

5.2.3 Active procedures pane

The information in the active procedures pane can be seen as an overview of the procedures being worked on, i.e. as a set of manuals on the desk. The purpose of this pane is to enable switching the attention between different activities in a simple way.

5.2.4 Overview pane

The overview pane is used for graphical display of procedure structures. At any time, the contents of the pane will be one of the following:

- a procedure graph
- an activity graph
- empty

5.2.5 Step survey pane

The step survey pane displays the instructions of procedure steps. All actual execution of procedure instructions is initiated from this pane.

The contents of the step survey pane will at any time be one of the following:

- instructions of the current step in the current activity
- instructions of a step being viewed
- empty

Each instruction in the step survey pane has an associated set of commands. The commands associated with every instruction are:

Command	Description
SKIP	Skip this instruction.
DO IT	Execute this instruction.
RECONSIDER	Go to a previous instruction.
COMMENT	Look at or enter a comment associated with this instruction.

Some instructions will have commands in addition to the ones above.

6. COPMA EXPERIMENTAL PROGRAM

In order to quantify performance improvements of reactor operators using COPMA compared to hard copy manual, an experimental program is set up in Halden Man Machine Laboratory.

6.1 First COPMA Exercise

The first COPMA validation exercise will be a limited-scope exercise disigned to test the mechanics of the COPMA system and its interface, and to provide useful information to incorporate into further enhancements of COPMA. In addition, the exercise will be used to provide an experience base for the full-scope COPMA-I experiment in the fall. The emphasis will be primarily to gain experience with the use of COPMA that will highlight its potential benefit as well as strong and weak points in its current implementation. The exercise is not intended to provide a statistically significant measuerement of an improvement in operator performance relative to any specific performance measure. However, the exercise will be performed in a formal manner so that the results can be published externally, and so the conclusions can be considered valid for purposes of system modifications.

The COPMA spring exercise will attempt to provide information to help answer the following questions:

1. Can procedures be computerised? Data and experience will be gathered during the adaptation of procedures to the COPMA structure and their installation into COPMA.

2. Can operators be trained to use a computerised procedures system? Data and experience will be collected during the COPMA training process. The training to use COPMA will be compared to the training to use the analogous hard-copy procedure.

3. Can computerised procedures be used to control a process? Is performance using COPMA better than that while using hard-copy procedures? Data and experience will be collected during the specific COPMA exercise scenario. Performance will be compared between the use of COPMA and the hard-copy procedures.

4. For what types of tasks does COPMA provide the greatest benefit? Performance using COPMA will be compared to performance using hard-copy procedures for different types of procedure following tasks.

Because of the limited scope of the first COPMA exercise, certain imporant questions will not be answered conclusively:

1. Does COPMA improve performance during disturbance situations?

2. Does COPMA facilitate (i.e. reduce time and errors) for transitions between procedures?

3. Does the use of COPMA improve the operator's understanding of the process and his location within the procedure system?

4. Does COPMA provide additional assistance when using a large set of procedures?

Although the first COPMA exercise will not answer these important questions, it will provide a good basis for the development of the fall experiment that will address these issues in more depth.

6.2 Full Scope COPMA Validation Experiment

1. Establish the value of the current version of COPMA. Identify and quantify the performance improvements discussed above by comparing operator performance with the COPMA system to performance with standard hard-copy procedures.

2. Provide feedback on the adequacy of the COPMA functions and the MMI to enhance further system development.

3. Qualitatively assess the value of computerised procedures by comparing the operator's role with COPMA to the role when using hard-copy procedures.

4. Provide information on requirements for integrating COPMA with other COSSs.

5. Provide feedback from test subjects regarding the value of COPMA and desired modifications.

6. Evaluate adequacy of methods and content of training program used to prepare test subjects.

7. CONCLUSION

A computerised procedure system has been developed and integrated in the advanced experimental control room facilities of the OECD Halden Reactor Project. The system will be experimentally validated through a series of experiments. Through these experiments one will to try to quantify the performance improvements comparing operator performance with the COPMA system to operator performance with standard hard-copy procedures.

Toward a Comprehensive System for Fault Diagnosis of Turbomachinery

PRADIP N. SHETH, DAVID W. LEWIS, and RAHUL NAHAR*
Mechanical and Aerospace Engineering Department
Rotating Machinery and Controls (ROMAC) Laboratory
University of Virginia
Charlottesville, Virginia 22901, USA

ABSTRACT

This paper describes a rule based system developed at the University of Virginia for vibration oriented diagnosis of turbomachinery for fault identification and for predictive maintenance. The system is implemented in a PC based PROLOG environment, with the Dempster Shafer theory of Belief Functions utilized for evidential support of hypotheses. The direct uses of PROLOG for knowledge representation, rule interpretation, control strategy, and user interaction are described.

The vibration fault diagnosis system is considered to be one component of a comprehensive system for turbomachinery. The framework of this comprehensive system comprises hierarchical levels of generic rules (surface knowledge) and generic analytical simulation models (deep knowledge). The root level includes the surface and deep knowledge for vibration, bearings, lubricants, seals, gears and couplings, and mechanical/metallurgical aspects of fault detection. Another level comprises the generic but specific knowledge base for various categories of turbomachinery, i.e., pumps, compressors, turbines, engines. The third level includes the installation specific rules, maintenance, repair, and troubleshooting logs, and other specific usage experiences. It is shown that each component of the comprehensive system can be viewed as a distinct expert system which can be developed and utilized independently of the other subsystems while the comprehensive system is evolved over a period of time.

INTRODUCTION

The process of diagnosing faults in turbomachinery is a multifaceted process. This process includes the use and consideration of such factors as (i) the experience knowledge base acquired from varied sources, e.g., the repair manuals, trouble-shooting handbooks, experienced consultants and local plant personnel, (ii) decisions on further exploratory test/analysis of subsystems to progressively narrow down the possible causes, and (iii) analysis and interpretation of sensor data, all augmented by the local perceptions of the plant personnel responsible for maintenance and repair of the specific turbomachine. The off-line troubleshooting

*Presently with ICM, Inc., Reston, VA

and preventive maintenance are the traditional approaches; however, the evolution of these off-line methods toward an on-line system which can serve as a predictive maintenance system in a significantly broader role than the traditional threshold based on-line alarms is a desirable goal. The application domain of both the off-line and the on-line diagnostic systems for turbomachinery includes the power generation systems, the chemical and process applications, a variety of vehicular power plants, sewage/water treatment facilities, minerals processing applications, manufacturing equipment, and others.

Over the past three years, the conceptual framework and a research prototype of a comprehensive system for turbomachinery fault diagnosis has been developed at the Rotating Machinery and Controls (ROMAC) Laboratory of the University of Virginia. The research prototype, called ROMEX (Rotating Machinery Expert System), is continually and progressively updated to reflect the current status of the research into the evolving framework for the specification of the comprehensive system. Comments, suggestions, and validation experiences of the ROMAC industrial partners are incorporated in the evolution of the specifications for the system and in the ROMEX prototype. The industrial partners of ROMAC currently number about 50 industrial companies including turbomachinery manufacturers, pump manufacturers, and the users of turbomachinery from the utilities and the process industries. ROMEX serves as a testbed for the conceptual framework and provides a vehicle for the validation of the concepts in actual industrial settings. In this paper the current status of both, the conceptual framework and ROMEX, are described.

COMPREHENSIVE SYSTEM CONCEPTUAL FRAMEWORK

Figure 1 shows an overall concept of a comprehensive diagnostics/maintenance system for turbomachinery. It is recognized that the diagnostic procedures, including the heuristic rules and the analytical/experimental modeling techniques, can be to a large degree broadly applicable to many types of rotating machinery if the focus is on such problems as misalignment, looseness, unbalance, improper meshing of gears, cracks, and resonance. This generic system can be at the root of a hierarchical system of diagnostics subsystems. A relatively smaller set of specific (but still generic) rule base and the associated modeling schemes can be utilized for the process specific equipment such as a pump, a turbine, or a compressor. Another set of specific (but still generic) rule base and the associated modeling schemes can be directed to components such as induction motors, synchronous motors, pivoted pad fluid film bearings, specific seal configurations, and other components. Such a component oriented view of a diagnostic system has resulted in a generic diagnostic system for manufacturing equipment [1]. Similarly, the installation specific rule base and the associated data base represent the relatively more specific information. Thus, the following hierarchy of genericity and thus a progressively expanding specificity is possible in a turbomachinery diagnostic system:

(1) Most Generic: Rules and models applicable to a broad class of rotating machinery; in many ways, these generic rules conceptually parallel the generic algorithms for such analytical tasks as rotor dynamic analysis which can be applied to a broad class of rotating machinery.

(2) Generic/specific to Process: Rules and models, while still generic, but specific for the process application such as pumps, compressors, turbines, fans, engines, etc. For example, performance related rules and models, and vibration excitations from aero/hydraulic forces would be specifically related to the specific process but generic enough for the entire class of process.

(3) Installation Specific Rules and Data Base: These involve the most specific rules and data base, i.e., specific to the plagued bearing, for example. The historical use, repair, and maintenance data for the specific installation would be a part of this data base.

Symptom-Cause Relationships

As indicated in Figure 1, the process of fault diagnosis in turbomachinery involves the interchangeable domains of symptoms and causes which manifest themselves as, for example, the vibratory behavior of the machine, the bearing performance such as the bearing temperature or power loss, lubricant data and contamination, seal behavior, and mechanical/ metallurgical observations of the components. The "symptoms" are those behavioral parameters which can be measured, analyzed, observed, or felt from the machine while the probable "causes" are inferred from these symptoms. The basic characteristics of these symptom-cause relationships are complicated by the following:

(a) There may be a number of symptoms about which the maintenance personnel would be uncertain because of incomplete/uncertain information from sensors;

(b) Different problems may have the same symptoms and also different symptoms may result from the same problem;

(c) The relationships form a hierarchical structure, requiring a progressively narrowing down search procedure as more evidence of symptoms-causes is generated.

As an illustration, the symptom of high "one per REV" (i.e., synchronous with rotor speed) vibration in a radial direction would point to the generic problem of unbalance. Unbalance in an operating machine can be caused by a variety of causes including a possible loss of a part, rotor bow, or a build-up on a rotating element. Each of these causes is quite different and, once identified correctly, requires a different search strategy. The diagnostic task can thus be divided into two major steps. The first step involves the diagnosis of a generic problem and the second step is the refinement and the progressive narrowing down from the preliminary diagnosis of the first step. This hierarchical structure for the first two levels is illustrated in Figure 2 for vibration based diagnostics of compressors.

Uncertainties in Data and Inexact Rules

The diagnostic system must be able to handle uncertainties in the data as well as the varying degree of beliefs in the various cause-symptom relationships established from a number of sources. Further, the assignment of higher or lower probabilities to specific problem causes, as appropriate, from the specific maintenance history of the machine under consideration is also a necessary requirement for the system. A variety of methods for handling uncertainties in diagnostic systems are available. The subjective Bayesian approach [2], the method of Certainty Factors [3], the fuzzy logic possibility theory [4], and the Dempster Shafer (DS) theory of belief functions [5] are some of the methods employed in diagnostics systems to quantify uncertainties. Of these methods, the DS theory of belief functions is particularly well suited to the turbomachinery fault diagnosis process because the progressive narrowing down of possible causes from evidentiary support is a fundamental task of the diagnostic process. For the comprehensive fault diagnostic system, the DS approach was selected, with the computational scheme proposed by Gordon and Shortliffe [6] employed to implement the DS scheme in ROMEX. The key benefits of the DS scheme include:

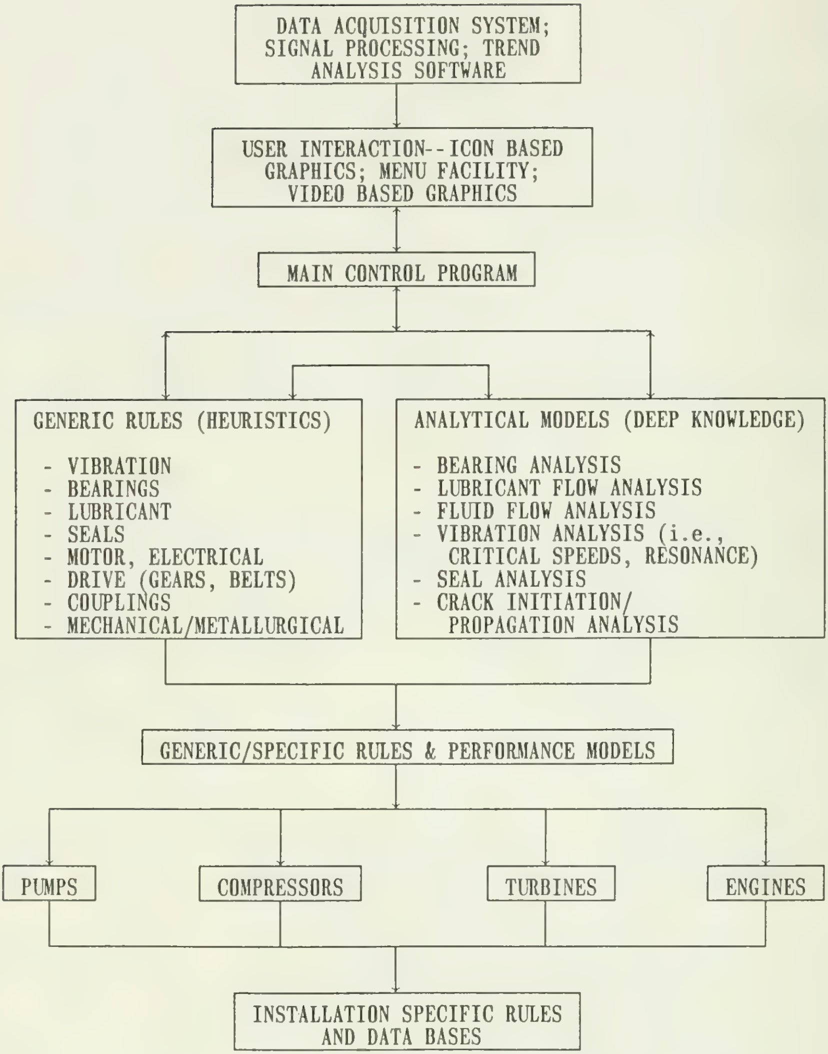


Figure 1. Conceptual Framework of a Comprehensive Diagnostics System for Turbomachinery

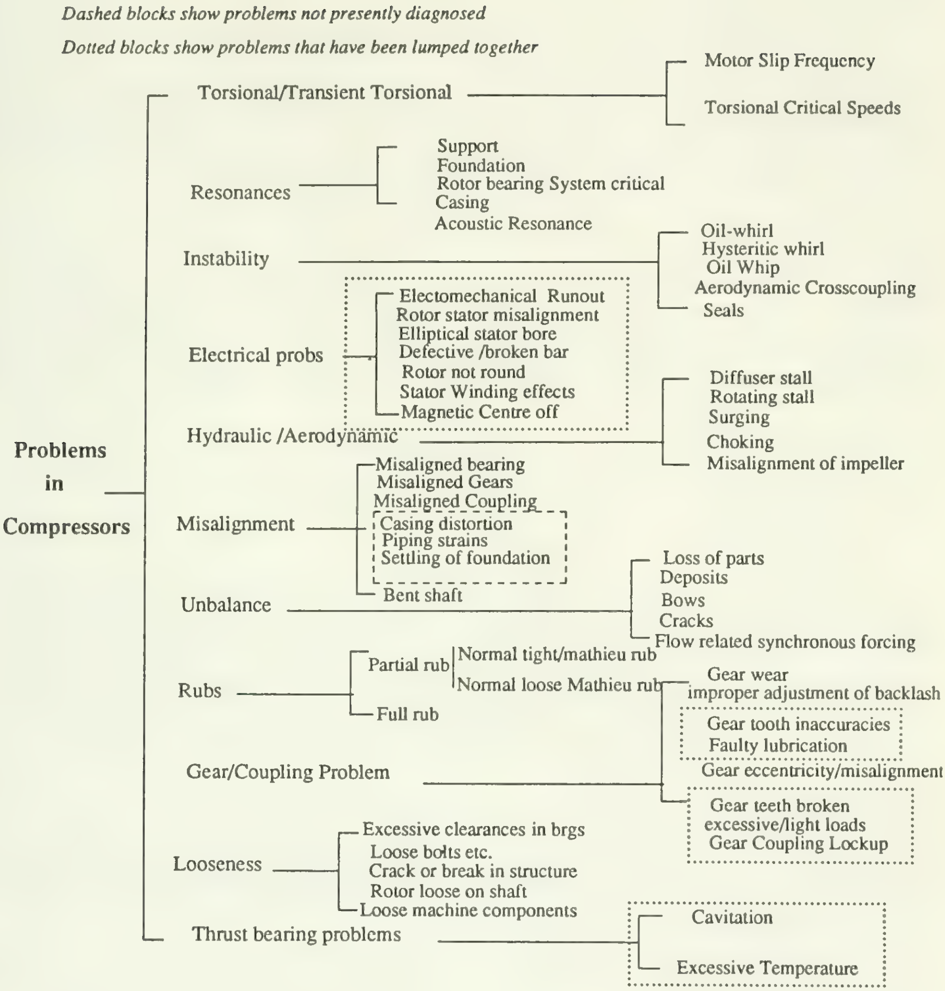


Figure 2. Hierarchical Structure of the Diagnostic System

(a) The DS scheme allows for managing uncertainty in a hierarchical decision space.

(b) The DS method allows inexact reasoning at whatever level of abstraction that is appropriate for the evidence that has been gathered at a particular stage in the diagnostic process.

(c) The DS model provides the ability to distinguish between lack of belief and equal belief.

The DS method asserts that the beliefs resulting from different evidences can be combined together only if the bodies of evidence are conceptually independent. This is a key assertion for the use of the DS theory and it is necessary to exercise cautionary judgment in utilizing the DS theory when quantifying the beliefs.

Data Acquisition

Besides visual observations or the feeling of unusual noise or other subjective parameters, quantitative sensory data are usually available for the diagnostic process. For vibratory performance alone a variety of probes would produce time histories and, in conjunction with an FFT analyzer system, would produce results in the frequency domain. Traditionally, various forms of data representation have been utilized for the diagnostics process, e.g.,

- Orbit plots
- Vibration spectrum
- Time histories
- Bode plots
- Cascade plots
- Polar plots, etc.

Each method of data representation is appropriate for one or more diagnostic search procedures. The diagnostic system can rely on the user to interpret the various data representations to submit the data required for the search procedure in response to a user query procedure. The interpretation of the data representations, which are performed off line, involves both one-to-one quantitative interpretation of sensory information and also a subjective assessment of the various spectra to assign the relative importance to the observed peaks and the rates of change in the various responses.

One alternative to the user interpretation of the sensory data is to incorporate a data acquisition and interpretation system which can directly interact with the fault diagnostic system and which can also interact and control the data acquisition process. Figure 3 shows a schematic representation of such an integrated approach for a vibration based diagnostic system. Several of the subsystems required for this integrated approach are "off the shelf", e.g., the spectrum analyzer and the IEEE-488 interface. The signal processing module, however, offers significant opportunities for innovative approaches to fault diagnostics. A neural net oriented method for pattern recognition, for example, can directly integrate the production rules of the fault diagnostic system thus combining the diagnostic system with the sensory data analysis in a single module. This would permit the implementation of some on-line capabilities to effect selected corrective actions resulting from the diagnostic process.

The facility for the user of the diagnostic system to define the specific layout and the input-output characteristics of the various sensors for the turbomachine of interest is a necessary ingredient for creating a machine data file. The available

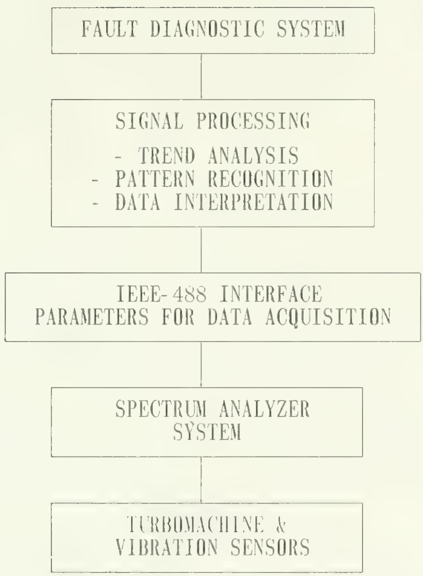


Figure 3. Integration of Vibration Data Processing with Fault Diagnostic System

sensors would necessarily dictate the course of the user query and that of the diagnostic search process.

Deep and Surface Models

The current generation diagnostic system is typically a collection of "pattern-action" rules which is intended to mimic the problem-solving heuristics of the expert. As discussed above, the hierarchical search structure and the mechanisms of evidentiary support for progressively firming up the degree of beliefs in various hypotheses provide a reasonable initial prototype for the turbomachinery fault diagnostic system. This may be characterized as a surface (or shallow) system. On the other end of the spectrum, a large number of algorithmic procedures are available — primarily in FORTRAN - for rotor/bearing system dynamic analyses, stability analyses, bearing analyses, flow analyses, and fluid/structure interactions, among others. These involve a variety of modeling, analytical, and experimental analysis techniques including the finite element techniques, modal analysis, and numerical methods for the time and frequency domain solutions. ROMAC, for example, has developed a library of over eighty (80) FORTRAN programs for turbomachinery analysis, which has been tested and validated over the past fifteen (15) years. These procedures, based on "first principles", may be designated deep models although there is no general definition yet on the form and the content of the deep models. Other possible types of deep models include: functional model [7] describing how a specific turbomachine works, detailed causal networks, and collection of rules of the form: if (symptom) then (cause) with (recommended action) which will produce (predicted response). This evolution of rules to include the predictive ability in the diagnostic system resulting from one or more recommended actions is one of several potential uses of the deep knowledge models. Within the framework of the comprehensive diagnostic system, the deep knowledge models are envisioned to complement the surface models in the following manner:

(a) Based on the design parameters, i.e., the structural, the mechanical, and the dynamic characteristics of the components of the turbomachine as installed at a specific site, the algorithmic procedures - including the analytical and the experimental techniques - can be utilized to establish a reference file of vibration and performance parameters. A tuned model of the turbomachine is then available to test the degree of beliefs in various hypotheses, in effect complementing the sensory data for the diagnostic process. Further, changes in the design parameters of the components over a period of time, if measurable, can be incorporated in the site model of the turbomachine. This concept of <u>model reference adaptive diagnostic system</u> is currently being evaluated as a part of the overall comprehensive system.

(b) The deep models can be utilized to create additional, or complementary, rules to the rules identified from expert knowledge. This idea of "compiled" deep knowledge has been utilized in a medical diagnostic system MDX [8]. In essence, the deep models are utilized at the knowledge acquisition stage to complement and perhaps validate the production rules acquired from experts.

(c) The deep models, as discussed, can be utilized as predictive tools for selecting and recommending appropriate corrective actions resulting from the diagnostic process.

Organization and Growth of Production Rule Base

As the comprehensive diagnostic system grows, the issues of discrepancies, ambiguities, redundancies, and completeness among the rules become critical. Also, a diagnostic system can never anticipate all of the potential symptom-cause

relationships at the development stage. An appropriate mechanism for modifying the rules already contained in the rule base, adding to the rule base, and for changing the quantitative belief functions for various hypotheses must be provided to facilitate the growth and "learning from usage" of the system. Rule checking procedures and programs have been developed for a medical system [9]. Another approach [10] covers additional problems in knowledge-base checking by considering unreachable and deadend clauses as well as circular rule chains. A decision table based approach is utilized in [11] to develop a general purpose Expert System Checker written in Pascal. ROMEX currently has about 80 production rules and the knowledge base is expanding. The use of an elementary rule checker in ROMEX is currently being tested.

Closely associated with the need for a rule checker to permit a cohesive learning growth of the diagnostic system is the need for a rule base editor. Such an editor would allow the user of the diagnostic system, perhaps the plant personnel, to (i) review the existing rule base, and (ii) add to the rule base with simple, English like inputs. The editor, in turn, would utilize the rule checker to ensure a cohesive growth of the rule base. Such an editor is currently being tested [12] within ROMEX.

Knowledge Acquisition and Validation

This is, of course, the most critical and probably the most difficult of all of the components which constitute the diagnostic system. The intent of the diagnostic system is to mimic the thought processes of an expert to arrive at conclusions regarding the probable causes (faults) of the observed and the measured symptoms. An obvious method for creating the knowledge base would be to work through a number of case histories of turbomachinery faults with one or more experts and hope that the experts are sufficiently prolific and the interviewer inquisitive enough to develop a probing description of the conscious and the sub-conscious thought processes involved in diagnosing a fault. It was realized, however, that the experts utilized for ROMEX were much more comfortable criticizing the rules and suggesting changes/new rules rather than starting from scratch and discussing how they go about diagnosing problems. The method utilized for knowledge acquisition and validation followed the following steps:

Step 1: A collection of rules for the initial knowledge base was developed from a variety of sources, including:

(i) Sawyer's Turbomachinery Maintenance Handbook (SOHRE Charts);
(ii) Case studies of turbomachinery diagnostics and problem solutions from

- ROMAC Conference Proceedings
- Texas A&M Turbomachinery Conference Proceedings
- Selected EPRI reports
- Interviews with the University of Virginia ROMAC faculty who are actively engaged in industrial consulting dealing with turbomachinery problems
- Selected journal articles and books on turbomachinery maintenance

Step 2: The compiled knowledge base of Step 1, reflected in production rules with appropriate belief functions, was incorporated in a diagnostic system.

Step 3: Selected case histories made available by a ROMAC industrial partner were diagnosed using the system. Initial results were encouraging; however, this is a continuing process for the refinement and the growth of the knowledge base. The case study based approach for the validation and refinement of the knowledge base has been successfully utilized before. For example, [13] reports the use of eight actual aircraft accident cases for the confirmation and refinement of a real-time fault diagnosis expert system for aircraft applications.

The knowledge base resulting from the above steps is described in [14]. The current knowledge base is concentrated on the vibration based diagnostic process. This task, in particular, is a continuing and iterative task in nature and the evolving comprehensive system framework is expected to be significantly shaped by the progress of the knowledge acquisition and validation task.

User/System Interaction

There are at least three aspects of the user/system interaction:

(i) The input and output dialog for defining a specific machine, its associated sensors, its repair and maintenance history, and other similar data. One possible appropriate method is to interface with one or more popular database managers such as, for example, DBASE III. The advantage of this method is that many plant personnel already utilize such systems for capturing the repair/maintenance history. For the mechanical and the structural design parameters, interface with CAD system databases would also be desirable.

(ii) For the user query to define symptom-cause relationships, the user must have the option of asking WHY? to a specific query or to a line of reasoning. Further, the use of still photographs of the components, photographs or CAD drawings of the electrical, hydraulic, piping, or other schematics should be utilized during the user query. The SA·VANT user interface system developed for the EXACT (Expert Advisor for Combustion Turbines) system is an example of the use of a video based graphics system in a diagnostic system [15, 16].

(iii) The user of the diagnostic system should have the option of reviewing the knowledge base by categories such as component faults, causes, symptoms, etc. The rule editor, described above, plays a vital role in this capability [12].

ROMEX PROTOTYPE

The current ROMEX prototype [14] is directed at vibration diagnostics and contains about eighty (80) production rules. The current rulebase contains the hierarchical structure for the following problem categories (see also Figure 2):

- Unbalance
- Mechanical looseness
- Misalignment
- Gear Problems
- Aerodynamic problems
- Coupling problems
- Thrust bearing problems
- Subharmonic resonances
- Harmonic resonances
- Some electrical problems

and, - Instability problems

Figure 4 shows the overall structure of the current diagnostic process.

A variety of expert system shells and aids are available commercially for quick prototyping efforts. A key consideration in the development of the research prototype has been the need for flexibility. A PC-based Prolog compiler, available commercially, provided the most suitable vehicle for the prototype development. Among the advantages of Prolog for the research prototype, the following are particularly prominent:

(a) Prolog provides a strong capability for pattern matching;

(b) Backward chaining inference engine is already built in the Prolog structure. The diagnostic system relies heavily on the backward chaining process. Also, other types of inference schemes can be relatively easily incorporated by utilizing the prolog facilities;

(c) With Prolog, the capabilities of the diagnostic system can be relatively more easily expanded and modified when compared to the use of a system development tool. Note that the framework for a comprehensive diagnostic system will continue to evolve and ROMEX will need to incorporate the necessary directions defined for the framework;

(d) Prolog also provides a relatively easy transportability of the system for testing at various industrial partners of ROMAC.

A meta-interpreter approach [17] was utilized in the Prolog language to implement the following:

- Mechanism for specifying certainties in rules and data;
- Mechanism for computing certainties of conclusion given the certainties of the rules and the premises;
- Mechanism for providing explanations.

The following is a brief description of the implementation issues:

Knowledge Representation

The rules as well as the facts are represented in ROMEX as prolog clauses. Each fact is represented as either an <object value> or <object attribute value> pair. For example: bearing(ball,inboard): here bearing is the object, inboard is the attribute, and ball is the value of the object. The fact bearing(inboard,ball) means that bearing located on the inboard side of the compressor is of the type ball-bearing. The knowledge about the truth of the fact is represented as a prolog clause "fclause/4". The first argument identifies the fact. The second argument of fclause states whether the fact is true or false or unknown (here "unknown" signifies that the user has been queried about the fact and he knows nothing whatsoever about the fact). The third argument to the fclause/4 gives the uncertainty in whether the fact is true or false. The fourth argument in fclause/4 gives the list of rules that were used to arrive at the fact. If it is a null list it implies that the fact was arrived at by querying the user.

An example of how a fact is represented is:

```
fclause(bearing(ball,inboard),true,1,[]):
```

This clause states that the inboard bearing is a ball bearing with the certainty 1 and this fact was established by querying the user. The rules are represented in the following manner:

```
check_clause(SupA,A,B,C,D):-

B = X,Y..Z  .
```

SupA is the super category A belongs to. A is the problem name/category. B is a collection of premises which need to be true for A to be true. X, Y and Z represent individual premise. Each premise could either be a:

- fact;
- negation of a fact;
- conjunction of a fact and premise;
- disjunction of a fact and premise;

Additionally, each premise could also be a prolog clause. Thus although the rule language is structured, full functionality of prolog is available to the user.

C denotes the certainty associated with the rule. D denotes the rule number. Thus the following rule number 15 in English:

if the predominant frequency is one times the running frequency and the amplitude is radial

then the initial problem is unbalance with certainty 0.5.

would be expressed as:

```
check_clause(iprob,unbalance,B,0.5):-

    B = pred_frequency(1),

        pred_ampli_dir(radial).
```

Note that the rule itself does not specify how the uncertainty is to be computed nor does it say how and when the facts have to be queried from the user. These tasks are accomplished by the rule-interpreter.

Rule Interpreter

The backward chaining interpreter is the primary inference mechanism in ROMEX. The prolog predicate **solve** is the basic mechanism for implementing this interpreter. It functions roughly as follows:

1) When given a goal, **solve** first checks whether it is already a fact in the memory. If the goal is found to be a fact in the memory then **solve** checks if the certainty associated with the fact is greater than 0.1 (minimum certainty threshold), if so then the goal is found to be true with the associated certainty else **solve** fails.

2) If the goal is not a fact in the memory **solve** checks whether there are rules which have the given goal as the consequent. If so, then it collects the premises of the rule and makes them its new goal. **Solve** succeeds if all premises are proved to be true and the combined certainty of the rules and each of the premises exceeds the threshold otherwise **solve** fails.

3) If the first two conditions are not true then **solve** checks whether the goal can be interpreted from the available facts. If so, it tries to interpret the value of the attribute to emulate "common sense" reasoning.

4) If the first three conditions do not apply then **solve** checks whether the question can be asked of the user inquiring about the truth/falsity of the fact. If so then the question is asked and users response is saved in the database. The user's response decides also whether the goal is true or false. If the goal is proved to be false **solve** fails.

Control Strategy

The rule-interpreter as described above, accomplishes the following functions. When given a goal it determines whether the goal is true, and the belief associated with the goal. There are a few other tasks which it also accomplishes. These are:

a) Formation of hypothesis.
b) Combining the uncertainty in the facts generated by more than one rule.
c) Displaying the results and providing explanations.

The control strategy consists of:

0) Make *Problem* as the top level node.

1) Form a hypothesis set comprising all of the subcategories of the current top level node.

2) For each hypothesis in the hypothesis set do the following steps:

   a) Use **solve** to prove the negation of the hypothesis. If **solve** returns the belief in the negation of the hypothesis as 1 then remove the hypothesis from the hypothesis set and stop, else continue.

   b) Use **solve** to prove the hypothesis. Repeat the process until all the rules relevant to the hypothesis are utilized. Collect the beliefs in the hypothesis generated by different rules and combine them to form a composite belief (step 1 of the DS approach).

3) The DS scheme is utilized (steps 1 and 2) over the current hypothesis set, and beliefs for and against each hypothesis are combined. The hypotheses are ranked in order of beliefs associated with them. The hypothesis with belief less than 0.1 are removed from the hypothesis set thus pruning the search space.

4) Investigate each hypothesis in the current hypothesis set by making it the top level node and going through step 0-3.

5) Finally all the beliefs in the various hypotheses that had been investigated are combined and the results are saved in the database.

6) The results are displayed and, if desired, explanations are provided.

System Operation

When the system starts it asks for the turbomachine's name. This information is used to check whether the information about the basic mechanical characteristics is

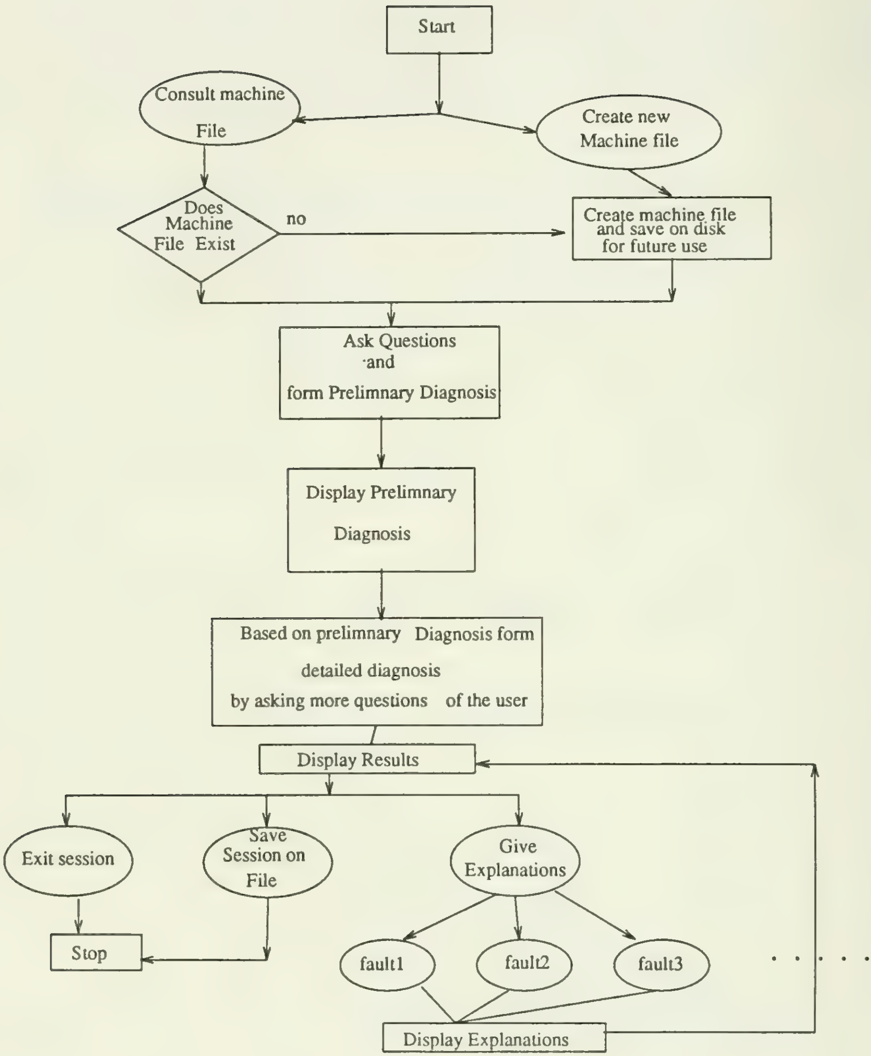


Figure 4. General System Architecture

stored on the disk. If it is then this information is retrieved from the file and added to the database, otherwise the user is queried about this information. The dialogue with the user is in the form of pop up menus. Once the mechanical characteristics are entered, the diagnostic process is started. Once the main problem categories are explored and the preliminary diagnosis arrived at, the results are displayed. Based on the preliminary diagnosis a detailed diagnosis is conducted when additional questions are asked of the user. When the final diagnosis is reached the result is displayed and the consultation process ends. At this stage explanations are given if desired. Explanation consists of displaying relevant rules required to reach the diagnosis. Figure 4 illustrates the basic flow of the diagnostic system. Entries in ellipses show the options available to the user.

CONCLUSIONS

The development of a fault diagnostic system for turbomachinery requires a resolution of a number of issues in expert systems, analytical/experimental methods, data acquisition systems, and user interaction. A two-pronged approach has been established for this development: (i) a research oriented facet of the development explores the various issues and concepts to establish an evolving, comprehensive framework for the diagnostic system, and (ii) a more pragmatic facet which implements the concepts in a working prototype called ROMEX such that a testbed for the comprehensive framework research is continually available for "hands-on" tests and validation. Industrial participation in evolving the framework and also in testing the prototype are necessarily key ingredients for the successful development of this system.

ACKNOWLEDGEMENTS

The financial support for this work was provided by the ROMAC Laboratory through its industrial partners, and by the Mechanical and Aerospace Engineering Department of the University of Virginia. The Center for Computer Aided Engineering provided the computer facilities for this project. Mike Buffer and Peter Accorti, graduate students currently working on this project, are acknowledged for their contributions.

REFERENCES

(1) Bajpai, A., "An Expert System Model for General-Purpose Diagnostics of Manufacturing Equipment," ASME Manufacturing Review, Vol. 1, No. 3, October 1988, pp. 180-187.

(2) Duda, R., Hart, P., and Nilsson, N., "Subjective Bayesian Methods in Rule Based Systems," Technical Note 124, SRI, 1984.

(3) Buchanan, B., and Shortliffe, E., Rule Based Expert Systems: The MYCIN Experiments, Addison-Wesley, Reading, MA, 1984.

(4) Zadeh, L. A., "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," Vol. 11, pp. 199-227, 1983.

(5) Shafer, G., A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ, 1976.

(6) Gordon, J., and Shortliffe, E., "A Method for Managing Evidential Reasoning in a Hierarchical Hypothesis Space," Proceedings of AAAI-84, Austin, TX, 1984.

(7) de Kleer, J., and Brown, J., "Mental Models of Physical Mechanisms and Their Acquisition," Cognitive Skills and Their Acquisition, Erlbaum Associates, Hillsdale, NJ, 1981, pp. 285-309.

(8) Chandrasekaran, B., and Mittal, S., "Deep Versus Compiled Knowledge Approaches to Diagnostic Problem-Solving," Int. J. Man-Machine Studies, Academic Press, Inc., Vol. 19, 1983, pp. 425-436.

(9) Suwa, M., Scott, A., and Shortliffe, E., "Completeness and Consistency in a Rule-Based System," Rule Based Expert Systems, Addison-Wesley Publishing Co., Reading, MA, 1984, pp. 159-170.

(10) Nguyen, T., Perkins, W., Laffey, T., and Pecora, D., "Checking an Expert Systems Knowledge Base for Consistency and Completeness," Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, CA, 1985, pp. 375-378.

(11) Cragun, B., and Steudel, H., "A Decision-Table-Based Processor for Checking Completeness and Consistency In Rule-Based Expert Systems," Int. J. Man-Machine Studies, Academic Press, Inc., Vol. 26, 1987, pp. 633-648.

(12) Buffer, M., M.S. Thesis In Progress, Mechanical & Aerospace Engineering, University of Virginia, 1989.

(13) Schutte, P., Abbott, K., Palmer, M., and Ricks, W., "An Evaluation of a Real-Time Fault Diagnosis Expert System for Aircraft Applications," 26th IEEE Conference on Decision and Control, December 9-11, 1987, Los Angeles, CA.

(14) Nahar, R., An Expert System For Vibration Fault Diagnosis of Turbomachinery, M.S. Thesis, Mechanical & Aerospace Engineering, University of Virginia, Charlottesville, January 1989.

(15) Koch, C., Isle, B., and Butler, A., "Intelligent User Interface for Expert Systems Applied to Gas Turbine Maintenance and Troubleshooting," Proceedings of the EPRI Conference on Expert Systems In Power Plants, May 27-29, 1987, Boston, MA.

(16) Shah, R., Pomeroy, B., Salemme, R., and Irving, R., "EXACT - An Expert System For Gas Turbine Troubleshooting," Proceedings of the EPRI Conference on Expert Systems In Power Plants, May 27-29, 1987, Boston, MA.

(17) Sterling, L., and Beer, R., "Incremental Flavor-Mixing of Meta-Interpreters for Expert System Construction," CH2340, IEEE, 1986.

Rapid Repair Advisor for Motor-Operated Valves: A Design Study

J. K. SOMSEL
Pacific Gas & Electric Company
Nuclear Power Generation Department
77 Beale Street Room 1435
San Francisco, California 94106, USA

ABSTRACT

The initial proposed application for the Rapid Repair Advisor project is for motor-operated valves (MOVs). The expected benefits from an MOV testing expert system depend on the purpose of the testing. Straight acceptance testing (post-maintenance and surveillance) could benefit from field verification of test validity. Troubleshooting of failed operators is seldom difficult. Intermittent problems are difficult to resolve suggesting that trace recording capabilities are needed. Predictive diagnosis places the most demands on the interpretive skills of the engineer. However, the limit to predictive capabilities seems to lie in the design of the MOV and the measurable parameters. Utilities are expected to require a knowledgeable MOV maintenance engineer to make decisions on MOV maintenance and operability. The economics of developing an expert system are comparable to improved training for the end-users.

PROJECT BACKGROUND

The Rapid Repair Advisor project is co-funded by Pacific Gas & Electric Company and the Electric Power Research Institute. It was initiated with the goal of providing immediate equipment diagnosis capabilities to the plant operators and technicians working backshifts and weekends. The rapidly evolving regulatory interest in motor-operated valves and the growing impact on nuclear plant costs prompted EPRI and PG&E to chose MOVs as their first target component.

To begin the project, a study of utility requirements was commissioned with PG&E serving as the project manager and prime contractor. This is phase I of the project with possible subsequent work depending on the results of phase I. The scope of phase I includes definition of the needs of utilities for MOV diagnostics, a survey of the current products and technologies for MOV testing, and recommendations for further work.

This report is based on the preliminary findings to date and does not represent the official conclusions of EPRI or PG&E. The views expressed here are solely those of the author.

INDUSTRY SITUATION AND MOTIVATIONS

The accumulating operating history for U.S. nuclear power plants points to shortcomings in motor-operated valve performance and reliability. The general perception from experiences in the fossil power and petrochemical industries is that MOVs are rugged and reliable. However, the demands for yet better availability under the more demanding conditions in nuclear plants has created a minor industry in MOV testing equipment.

The demands for improved performance stem from a number of specific cases where MOV failures have been prime contributors to significant nuclear power plant incidents. The most famous of these was the Davis-Besse event of 1985 where the failure of one MOV to open greatly increased the risk significance of a relatively minor initiating event.

The result of the Davis-Besse incident and its regulatory aftermath was the adoption of formal valve testing programs at most plants. Further regulatory requirements are in the works that will demand even more extensive testing.

Utilities recognize the problems with MOVs and seek to improve MOV performance. At the same time, the testing activities can generate significant radiation exposures for plant workers that need to be minimized. The dollar cost of the testing programs is also growing rapidly.

Another problem many utilities are experiencing is that most testing and preventive maintenance activities require the plant to be shutdown. This is putting more and more activity into the refueling outages and threatening to extend the outage duration. The cost impact from extending an outage is very great.

We can summarize the interests of the utilities into three goals; 1) improve valve reliability, 2) control costs, and 3) reduce man-rem expended. Minimizing the impact of valve work on outages can also be a significant in many cases.

TESTING GOALS

In support of the general interests of utilities discussed above, plants can test MOVs using three general tactics. The focus of each individual plant's program will vary given its age or maturity and the number and role of the installed MOVs.

The most common current use of MOV testing equipment is in acceptance testing. This could be either post-maintenance testing or surveillance testing. The acceptance criterion here is whether or not the operator delivers the thrust as prescribed by the vendor calculations. These calculations are performed to estimate the closing or opening torque required for a particular valve under worst case operating conditions. This value is translated into torque switch settings that are calibrated using a baseline load cell test. These settings are regularly verified to assure that the torque switch settings have not drifted.

Acceptance testing is generally a go/no go effort. A preventive maintenance program is conducted concurrently for such items as stem lubrication, grease changing, and general refurbishment. The intervals for the preventative maintenance activities are set based on predetermined intervals.

Troubleshooting or diagnosis of MOVs known to be inoperable did not seem to be of great importance to most utilities we surveyed. It was generally recognized that knowing the specific cause of a failed MOV before repair was unimportant because the range of maintenance responses was narrow. The big decision is whether or not to open up the case of the MOV. Once open, the cause was usually immediately apparent. The diagnostic process to make that decision was not so complicated to require computer assistance.

An area where new computer support was indicated was in the diagnosis of "gremlins" or intermittent troubles. A common occurrence was for the Operations department to report that a problem had occurred on the backshift. When maintenance crew arrived the next morning to repair the problem, no fault could be found. The ability to instrument an MOV and record the traces from all strokes including during actual demands appears to be of value.

Most plants we talked to also had "problem children" or MOVs that had subtle or evasive problems that were difficult to pin down. The new plants with only a few years of operation had the most. These were usually attributed to design problems, misapplication, or difficult operating environments. Diagnosis of these kinds of problems is a difficult task, challenging to even expert engineers.

The ability to perform predictive diagnosis would find wide support. Ideally, a utility would like to do needed preventive maintenance at its convenience but before the valve would otherwise fail. The current practise is to establish a set of chronological guidelines that will conservatively initiate maintenance before it is required. These preventative maintenance activities are based on industry, plant, and vendor experience.

Another technique that can be applied to initiating preventive maintenance is condition monitoring. Here, measurable parameters are identified and used as signals to initiate maintenance. One that is used at most plants is grease sampling to indicate when it is time to change the grease. Predictive diagnosis and condition monitoring are similar concepts but differ in that predictive diagnosis should identified unexpected faults. Condition monitoring only follows expected degradations.

TESTING METHODS

A number of companies are marketing equipment to assist in the testing of MOVs. The earliest system to market and the holder of about 60% of the market share is the MOVATS system. Other companies have followed offering either refinements to the underlying MOVATS techniques or new measurement technology. Systems are offered as of this writing by MOVATS, Westinghouse (VITAL), Liberty Technologies (VOTES), Impell (OATIS), and Wyle (V-MODS).

The critical difference between these systems is the parameters measured. An MOV is a device that converts electrical current into valve stem thrust in a controlled manner. The electrical current is started from an external command by closing the power supply breaker. Current is switched off by either stem limit switches or torque switches. The torque switches are very

important components as they typically determine how tight the valve will shut. Should they be set too low, the operator may not deliver enough thrust to even move the stem. Set too high, the operator could destroy itself and the valve. The largest cause of MOV inoperability has been incorrect torque switch settings.

To determine operability, the prime question has been, can the operator deliver the required thrust? A seemingly simple question but one difficult to answer. A direct measurement of thrust requires that the strain in the valve stem be measured. The stem, however, moves up and down or else rotates. Stem movement usually precludes the attachment of strain gages directly to the stem. A load cell can be mounted atop of the operator to measure opening thrust only; closing thrust is usually as or more important.

The older MOVATS system measures an internal parameter called spring pack displacement that can be roughly correlated to the delivered thrust. However, significant controversy exists as to how much this measurement is to be trusted. The spring pack itself is a common source of problems and uncertainties and is located only midway in the power path from motor to stem. However, the MOVATS system has the most field experience and many satisfied users. Another system, OATIS by Impell, also used spring pack displacement as the prime test parameter but has been partially withdrawn from the market.

Two newer systems have enter the market, VOTES and VITAL, that infer stem thrust by measuring its reaction through the valve yoke. A strain gage is attached to the yoke; any force developed by the stem and transferred to the valve must, by Newton's law, develop an equal and opposite force that is delivered via the yoke.

Another system is under development by Wyle that has yet to be released to the market. It will use sophisticated signal analysis of the motor current phase angles to detect changes from a baseline calibration stroke of the valve. The payoff here is that subsequent testing can be performed from the motor control center that supplies power to the valve motor rather than at the valve.

Once the signals are measured, most systems use a portable computer to collect, process, digitize and record the data. The data is reduced to traces of the parameters versus time. The traces can be further analyzed to extract pertinent features from the traces such as time of peak current or steady running current. MOVATS uses a recording oscilloscope that can store the collected data in bubble memory. Conversion to DOS-compatible format is available. The extracted feature tables would be the raw materials for an expert system.

ANTICIPATED ADVANTAGES

Engineers and technicians at the plants would ideally like for a machine to interpret the traces for them and tell them 1) whether or not the test just performed was valid so they can disassemble the test gear, 2) whether or not the MOV delivered its required thrust in both directions, and 3) whether or not the valve is going to perform as required until the next test. The first is relatively easy, the second is straightforward but not too trustworthy, and the third is difficult even for the best human experts.

The above wish list is really a list of decisions that must be made. Right now, the technician performs the test and the engineer interprets the trace and makes decisions. Can the plant management allow the technician aided by an expert system make the decisions as to operability and maintenance that engineers now make? The answer at almost all the plants we talked with is that the technician, usually union people, would not be allowed to make such decisions.

LIMITATIONS

The primary limitation to predictive maintenance is in the testing systems now on the market. The systems reviewed offered valuable assistance in acceptance testing but had only limited use, even in the hands of an experience user, for predictive diagnosis. The NRC published a list of 30 failure modes for MOVs in a draft Generic Letter on MOV testing. Almost all of these internal to a MOV could be detected by the more sophisticated testing systems according to our survey of practitioners. However, the ability to detect the onset of these problems in time for preventative maintenance was not established.

Based on our experience and conversations with other plants, every plant will need an expert engineer (or one in training) to manage the program and provide tender loving care to the MOVs. Some one engineer will need to be responsible for MOV performance. This also means trace interpretation.

The industrial organization of all U.S. plants we know of will dictate that that person will be an engineer and not a union member. This is not to say that a trained technician is unable to interpret traces or that a college degree is essential; only that the authority is not delegated to the technician. One concern of management is that a union member has a conflict of interest; deciding that more MOV maintenance is needed means more work for union members.

ECONOMICS

There are roughly 65 nuclear power plants in the U.S. with a total of 110 reactors. Given that one or two engineers at each plant site are responsible for MOV trace interpretation and that each plant has a testing program, only about 100 users are expected. The estimated cost for developing a predictive diagnosis expert system is $500,000 to $750,000. Additional costs would be incurred in distribution and maintenance. This means that the industry would spend over $5,000 per MOV engineer to develop the expert system.

A change in policy by the utility management to delegate maintenance and operability decision-making to technicians would broaden the user base. However, the number of users would not get much larger than the number of fielded testing systems or perhaps 200. The economics are not sensitive to the decision-making structure issue except that a technician user will require a more robust expert system.

A commercial problem foreseen is that the various testing systems on the market do not collect the same data and do not store it in the same format. Utilities will expect that all the systems be supported which means that at least two different knowledge bases will be developed (MOVATS/OATIS and

VOTES/VITAL and perhaps Wyle V-MODS). Cooperation from the vendors may be required to allow stability in data formats. A vendor could deliberately inhibit the use of an industry-developed expert system with its testing system by changing his data format to one incompatible with the industry expert system.

CONCLUSIONS AND RECOMMENDATIONS

The preliminary conclusion of this investigator is that expert system development for MOV diagnosis is not recommended. This is based on the following:

1) no great improvement in MOV diagnosis capabilities over trained human users is foreseen due to limits in MOV and testing technologies
2) the cost per user is high, other alternatives like training appear more cost effective
3) utilities will continue to require trained engineers to make decisions; giving less skilled technicians an expert system will not change that policy.

Development of an On-Line Expert System: Heat Rate Degradation Expert System Advisor

DALE M. SOPOCY and RAY E. HENRY
Sargent & Lundy

STEPHEN M. GEHL and S. MURTHY DIVAKARUNI
Electric Power Research Institute
3412 Hillview Avenue
Palo Alto, California 94304, USA

ABSTRACT

An on-line expert system for fossil-fueled power plants, the "Heat Rate Degradation Expert System Advisor," is being developed. This expert system will operate on a microcomputer and will interface with existing plant data acquisition and/or thermal performance monitoring systems, which presents a variety of design challenges. Several significant features of this expert system are discussed in this paper, including the use of a Project Application Specification, design of a modular expert system architecture, design of a consistent user interface, development of a standard specification for information transfer, and application of sensor validation techniques.

INTRODUCTION

Power plant operation has changed over the last 10 to 15 years, requiring many units to be operated in a cycling mode, placing new stresses on components, and having a negative impact on plant heat rate. Rising fuel costs, on the other hand, are causing additional emphasis to be placed on plant efficiency. The advancing age of existing units, which brings increasing mean-time-to-repair and maintenance costs for plant components, compounds the problems presented by these changes and can thereby hamper reliable, low-cost plant operation. Furthermore, environmental concerns are causing requirements for plants to be more tolerant of changing fuel characteristics while achieving lower emissions, which in turn creates a need to monitor and control plant operating parameters within tighter constraints. Consequently, utilities must implement measures to mitigate these factors at existing fossil-fueled power plants.

Heat rate is a major factor in the overall performance of a power plant. Therefore, measures that can facilitate operator detection and identification of the source(s) of any degradation in plant heat rate and speed operator response in taking the best overall corrective action are highly desirable.

Historically, most utilities have monitored heat rate monthly by the uncorrected ratio of total fuel consumption to total gross generation (accounting heat rate), at a minimum. This practice has been primarily useful as a crude measure of relative operating costs. This minimal level of heat rate monitoring does not provide either plant operators or performance engineers with any useful information pertinent to improving methods of plant operation throughout changing operating conditions, but rather with a monthly "report card" on the combined overall effectiveness of all plant operations.

Recently, some power plants have installed on-line performance monitors which provide instantaneous output on unit heat rate, as well as the actual and target values of various controllable parameters. While these monitoring systems can provide a wealth of detailed numerical information, and graphical comparison and trend charts that provide an indication of unit performance, they do not identify either the root cause or causes of off-design performance nor do they suggest any remedial measures. Given a charter to generate electricity at the lowest possible cost while maintaining the highest possible availability, plant operators are literally being placed in a position to fail by being provided with too much simultaneous information from on-line performance monitors and other plant monitoring systems to be able to assimilate and interpret fully all available information.

The next logical step in the evolution of plant performance control is an expert system that will both diagnose performance data provided by on-line performance monitors and assist plant operators that lack a full complement of on-line instrumentation and performance monitors in evaluating and diagnosing plant performance. Recognizing an industry-wide need for this advanced capability, the Electric Power Research Institute (EPRI) has undertaken the development and demonstration of an on-line expert system called "Heat Rate Degradation Expert System Advisor." This expert system will enhance the logic trees previously developed and documented in EPRI Report CS-4554, "Heat Rate Improvement Guidelines for Existing Fossil Plants" (1), with analytical relationships and supplemental heuristic, or practical, knowledge to enable interpretation of on-line data, and will automate the application of these enhanced guidelines using available on-line plant physical and

performance data and manual input of additional measurements and observations when necessary.

The "Heat Rate Degradation Expert System Advisor" will assist plant operating personnel in more closely monitoring plant heat rate, evaluating more of the available information, and evaluating more of the possible courses of action that will enable more responsive control of the parameters and equipment that affect plant heat rate than would be practicable with conventional performance monitoring alone. This expert system will also help heighten pla t operating personnel awareness of the developing conditions of plant equipment. The primary goals of this expert system are to enable each utility user to accomplish a measurable improvement in plant heat rate through improved response to both major and minor changes in plant operating conditions, while providing sufficient flexibility in design to facilitate widespread, rapid implementation throughout the utility industry. Secondary goals of this expert system are to enable utility users to increase unit generation capability, when applicable, and to increase equipment reliability and unit availability by focusing attention on the connections among plant operation, equipment life, and long-term performance.

Since the expert system will be drawing conclusions from both on-line and off-line instrument measurements, sensor validation routines are being developed and included in this expert system so that all diagnoses reflect the probable validity of the available data.

"Heat Rate Degradation Expert System Advisor" development is being divided into two consecutive phases over a four-year period as follows:

- Phase I - Development and On-Line Industry Demonstration of Expert System (1989-1992)
- Phase II - Commercialization of Expert System (1992-1993)

OBJECTIVES

The "Heat Rate Degradation Expert System Advisor" will provide on-line diagnosis of heat rate degradation that will help improve plant heat rate, availability, and equipment life by

- assisting operators in determining the best operating choices for operating at the "best achievable" heat rate, thereby reducing fuel costs; and
- assisting plant performance engineers in identifying subtle trends in plant performance that otherwise would be masked by variations in

operating conditions, differentiating between significant and insignificant (statistically uncertain) trends, and helping trouble-shoot degradations in plant performance, thereby increasing knowledge of the developing conditions of plant equipment.

This additional information may be used to schedule outages more effectively, as well as to make adjustments to operations to avoid forced outages, to improve equipment reliability, and, ultimately, to increase plant availability.

This expert system will accomplish these goals by meeting the following objectives:

- Identify meaningful short- and long-term trends in heat rate and in those parameters that affect heat rate, and recommend actions for further checking of off-design conditions.
- Diagnose meaningful deviations of heat rate and controllable parameters, and provide practical and reliable advice for identifying probable causes of performance degradation.
- Identify possible corrective actions for probable causes of performance degradation, and provide options for improving and/or optimizing plant performance.

The "Heat Rate Degradation Expert System Advisor" will be applicable to the majority of fossil-fueled power plants, including those having

- coal, oil, natural gas, and lignite fuels,
- reheat and non-reheat cycles, and
- subcritical and supercritical cycles.

PROJECT TEAM

The "Heat Rate Degradation Expert System Advisor" is being developed for EPRI by Sargent & Lundy. Sargent & Lundy is supplementing their in-house expertise in heat rate and expert systems development with several subcontractors, including Power Technologies, Incorporated (heat rate diagnostics), TRAX Corporation (modeling), and the University of California at Berkeley (sensor validation). The project team also includes Domain Experts and selected plant operations personnel from Commonwealth Edison Company and Northern Indiana Public Service Company that will be providing additional analytical and heuristic relationships for the knowledge base of the expert system. The involvement of the Host Utilities is an important aspect of the expert system development process and includes

- providing input to the design features and functions of the expert system;

- identifying needed revisions and/or additions to the analytical and heuristic relationships provided by the existing knowledge base on heat rate degradation;
- developing needed additional analytical and heuristic relationships;
- testing incremental versions of the expert system throughout development;
- field testing of the prototype and prerelease versions of the expert system; and
- reviewing validation and verification of completed expert system.

The completed, prerelease version of the expert system will also be field tested by a Demonstration Utility, Southern California Edison Company.

EXPERT SYSTEM DEVELOPMENT

Project Application Specification

A Project Application Specification is being used to guide the development of the expert system that specifies the functionality of every intended feature and describes the appearance and function of each intended type of display screen of the expert system.

Some of these features that are being provided in "Heat Rate Degradation Expert System Advisor" include

- the capability to be configured by users to individual plants without code customization through the use of menus and data input forms for the input of key unit design data and operating parameters that are referenced in the knowledge base;
- the capability to accept input and display results in either English or International System units;
- the capability to make use of appropriate on-line data from existing plant performance monitoring systems, data acquisition systems, data highways, and/or data loggers, provided that these data are made available to the expert system in files conforming to a specified format;
- the capability to continue functioning when some critical instrumentation is out of service by allowing manual input from the user;
- the capability to function off-line, without a performance monitoring system or other source of on-line data, by allowing manual input from the user;
- the capability to exchange data with separate boiler and turbine modeling programs, providing that data are made available to the expert system in files conforming to a specified format;

- the capability to use relevant results generated by other expert and/or diagnostic systems, if made available in the appropriate format;

- the capability to use input files that have been appropriately created or modified by the user to evaluate "what if" scenarios.

- the incorporation of sensor validation techniques to provide estimates of the "true" value and precision and an assessment of the probable validity of on-line input data, the provision of a trend analysis, and the identification of meaningful (statistically certain) trends;

- the capability to identify and diagnose meaningful deviations of heat rate and controllable parameters and to suggest remedial action(s) for correction;

- the capability to assess the certainty of each possible conclusion that is reached and to provide an explanation of the approach and reasons used to reach that conclusion; and

- the provision of interactive output on a CRT and the printing of (or storing of on a disk file) summary reports of input data, results, and explanations.

Expert System Architecture Design

The expert system architecture is being designed to provide a balance among efficiency of development and modification, efficient utilization of computer memory, adequate execution speed to provide meaningful support to plant operations, and self-contained consistency of user interface and explanatory capabilities. The basic components of this architecture include the major structural elements of the expert system (e.g., knowledge base(s), inference engine, user interface, and sensor validation routines), all appropriate input/output sources and destinations external to the expert system (e.g., thermal performance monitoring system, heat balance model, plant computer, data acquisition system, and keyboard input), and all required input/output interfaces. A simplified diagram of the "Heat Rate Degradation Expert System Advisor," illustrating the flow of information among major components, is provided in Figure 1.

A knowledge base architecture is being designed that will facilitate the formation of a knowledge base that uses computer memory efficiently, executes promptly, and can be easily modified during development. Other factors that are being considered in the development of the knowledge base architecture include organization according to the natural hierarchies of the knowledge, expression of the degree of uncertainty in the knowledge, organization to provide modularity of knowledge, performance

requirements, size of interfacing database, explanatory requirements, and the completeness of available information.

The design will provide for modularity, where appropriate, and set forth a specific methodology to be used at a global level, if practicable, to accomplish the following, in the order given:

- Check for presence of information
- Use available on-line data
- Use available results from other expert systems
  - confirm/refute conclusions
  - initiate consultations
- Request manual input
  - when insufficient information is available
  - if further definition of conclusions is desired

This design approach, incorporated into the design of the knowledge base architecture, will accomplish four objectives that are critical to the degree of use that will be attained by the completed expert system:

- The minimization of manual user input.
- The ability to operate with varying amounts of on-line instrumentation, generally increasing the certainty of the conclusions as the amount of available information increases.
- The ability to continue functioning when key instrumentation is not functioning, by requesting manual input from the user.
- The ability to make use of relevant results generated by other expert systems, if made available in the appropriate format.

User Interface Design

The expert system user interface is being designed in accordance with the EPRIGEMS™ Product Specification (2). The expert system's user interface is also being designed with an emphasis on ease of use by the primary user, the plant operator. In particular, the expert system is being designed to operate without attention from the plant operator until a significant condition is identified by the expert system, given that a minimal level of on-line instrumentation, sufficient to determine key measures of performance, is available and has been properly interfaced to the expert system. The expert system will also operate upon manual initiation whenever the

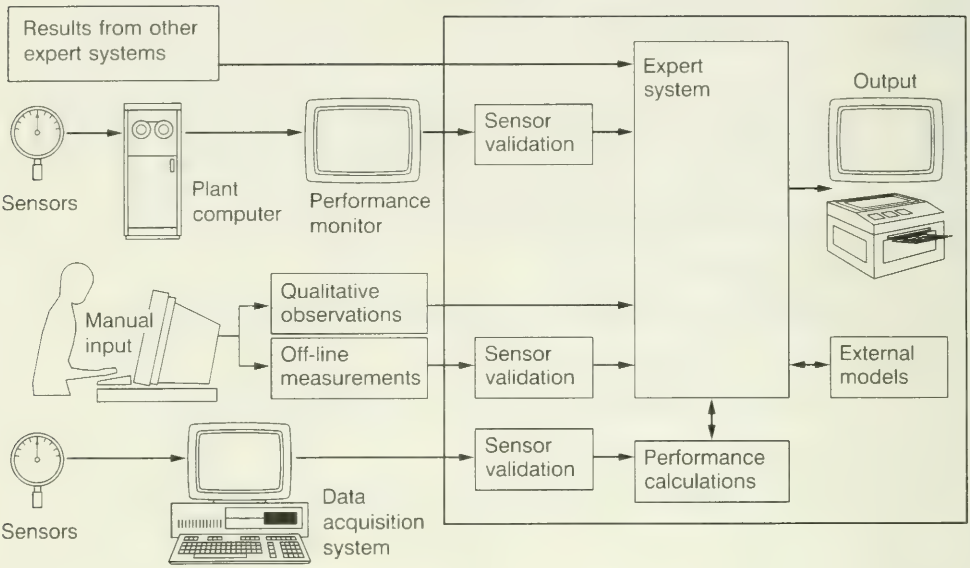


Figure 1—Basic Information Flow Diagram for Heat Rate Degradation Expert System Advisor

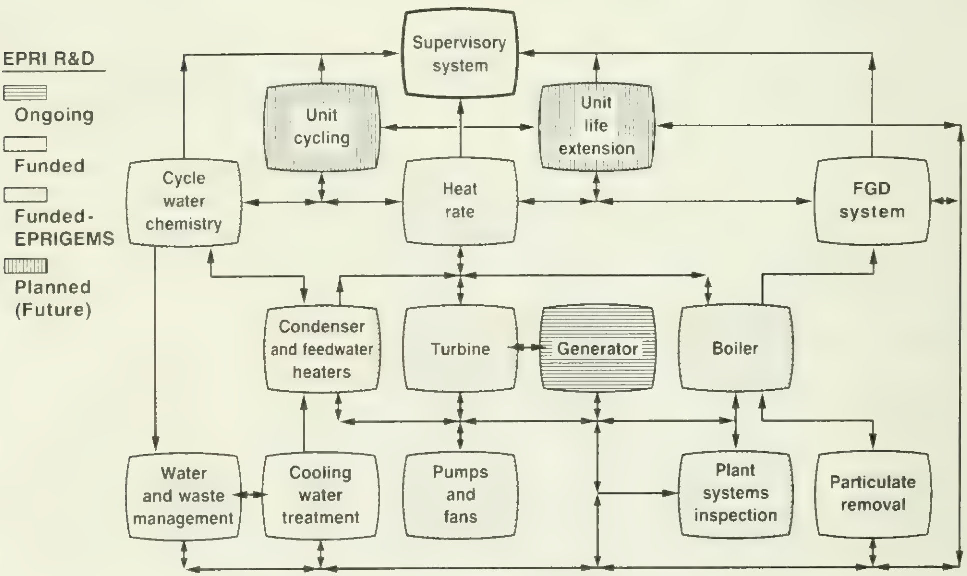


Figure 2—Prospective Expert Systems Network for Fossil Fueled Power Plants

operator desires to review and diagnose plant performance, whether or not any on-line instrumentation, sufficient to determine key measures of performance, is available to the expert system.

The expert system is being provided with a main screen display that will present a graphics-oriented summary of plant performance that will give a simple, visual indication of any significant deviations. Elements of subsequent screens will include menus, graphics of individual components and systems, graphic illustrations of identified trends, text windows, and tabular data, when appropriate. The user will be able to access additional screens that identify the input data and logic used by the expert system to diagnose a particular condition. Screens are being designed to facilitate intuitive navigation by the user to desired functions throughout the expert system.

Prompts for multiple manual user entries of numerical data are being presented in tabular or spreadsheet form for ease of entry, whenever possible. Prompts for manual entry of non-numerical user input are being accompanied by "point and shoot," multiple-choice response menus whenever practicable to minimize user typing and avoid ambiguity in user responses.

DESIGN CONSIDERATIONS FOR INTEGRATED USE OF ON-LINE DATA

Communication Interface and File Transfer Specifications for Information Transfer

Successful development and implementation of expert systems has been proven to be dependent upon limiting the domain of an expert system to a relatively narrow, well-defined application. Therefore, numerous expert system applications can be expected to be developed over the next several years for fossil fuel power plant operations (3). Some of these expert systems will address specific process aspects of plant operations, such as "Heat Rate Degradation Expert System Advisor." Other expert systems will address specific equipment, such as "Electrical Generator Monitoring System," "Turbine Condition Monitoring System," and "Condenser and Feedwater Heater Advisor" (4). Figure 2 illustrates all of the individual expert system applications for fossil fuel power plant operations that have been developed, are being developed, or are planned for future development at this time, including both EPRI- and independently-developed expert systems (5). Should all of these expert systems be developed without any capabilities for communication or integration, they will aggravate the burgeoning problem of "islands of information" faced by power plants with the proliferation of microprocessor-based local controls and the use of personal computers for a wide range of tasks (6).

The "Heat Rate Degradation Expert System Advisor" is being developed with the capabilities to export information and results that could be of use to other expert systems for plant operations. Similar capabilities are being provided that will enable this expert system to make use of relevant results that might be available from other expert systems. The anticipated paths of information flow between individual expert system applications are illustrated in Figure 2.

These capabilities are being accomplished through the design of the expert system and knowledge base architectures and the specification of a software interface design for the expert system that treats suitably conditioned on-line data and results as data elements in a database for access and interpretation by the expert system (7). This approach is also being used to enable the "Heat Rate Degradation Expert System Advisor" to function with a wide range of on-line data acquisition and/or thermal performance monitoring systems, as well as any related expert systems. This specification is intended to establish an "open architecture" that should facilitate, and thereby stimulate, integration of "Heat Rate Degradation Expert System Advisor" with both existing and future plant systems.

The communication interface specification that is being prepared will provide guidance to utilities preparing to install the expert system, both for determining any hardware interface requirements and for developing a file conversion utility for the import of data from commercial data acquisition systems, performance monitoring systems, and/or plant computers to the expert system. This specification will identify all parameters that are referenced in the expert system knowledge base and have the possibility of being available on-line, and will specify the information required by the expert system on each of these parameters, including value and data sampling rate. The form and format by which this information is to be stored for access and use by the expert system will be defined in this specification. The communication interface specification will be made available to other expert system developers to encourage development of compatible import and export capabilities that will enable integration of these systems with the "Heat Rate Degradation Expert System Advisor."

The communication interface specification is being developed with consideration of the capabilities and limitations of systems currently in use so that users can adapt existing equipment and software to the requirements of the interface with minimal effort and expense. The extent of this user effort is expected to be the development of a file conversion utility to translate the existing form of the available

data to the form and format specified for the application and in some cases the installation of some type of hardware interface.

The communication interface specification will include file specifications for the exchange of information between the expert system and each type of external, computer-based source or destination of information identified in the design expert system architecture. These specifications will include a standardized file format designed to facilitate the exchange of information between this expert system and existing and future stand-alone plant systems, including the plant computer as well as related expert system applications and commercially available models.

The file specifications will also include a standard vocabulary of variable names for all information expected to be input to, or derived within, the expert system knowledge base. This vocabulary will be used throughout the development of the knowledge base and will also be used to describe any information that may be exported from the expert system in the information transfer file. Similarly, a separate standard vocabulary that describes all variable states, or conditions, expected to be determined by the expert system for the range of identified variables will also be included in these specifications.

Sensor Validation

Any expert system that uses measured data as a source of information, whether input off-line as discrete values or on-line as continuous or semi-continuous values, must address the uncertainty of these measurements. Interpretation of sensor-based measurements is not always straightforward. Process trends need to be identified with a high sensitivity for the potential of an on-line expert system to be fully realized, yet sensor-caused inconsistencies in the available data must be identified correctly and not interpreted as process trends. Measured values may reflect background electronic "noise," instrument drift, and calibration errors in addition to the "true" value of the parameter being measured. Consequently, the precision and bias of each type of instrument is needed to quantify the relationship between measured and probable "true" values (8). However, this information alone is not sufficient to interpret measured data accurately. The possibility of instrument malfunctions must also be addressed. Furthermore, the response characteristics of some instruments and/or measurement techniques can impose a time-dependent element on the correlation between measured and "true" values.

Sensor validation techniques are being developed that will address these inherent problems with measured data while determining the validity of on-line data used by

the "Heat Rate Degradation Expert System Advisor." These techniques include domain-specific relationships for making consistency checks between related pieces of data. These consistency checks, which will be specific to this expert system application, will be supplemented with other, less application-specific, sensor validation techniques, including:

- Signal Feature Identification and Extraction (9): Domain-specific critical features that may be identified from the available on-line data will be classified according to time dependency and relative importance. This information will be mapped to symbolic values and supplied to the expert system for interpretation. A framework for integrating on-line data processing and the expert system will also be developed and implemented.

- Sensor-Based Management of Uncertainty: An architecture will be developed and implemented to manage the uncertainty of sensor information on a real-time basis. This architecture will incorporate in a systematic and efficient manner methods to fuse information from multiple sensors that augment traditional statistical methods of sensor validation. In addition, a methodology will be developed and implemented to update the uncertainty estimates associated with various components based on historical and current information, if practicable.

Separate modules of computer code are being developed for the "Heat Rate Degradation Expert System Advisor" to validate sensor measurements, provide trend analysis, and identify trends that are statistically certain. These separate sensor validation modules will be identified, and their intended functions defined, in the Project Application Specification.

General sensor characteristics used for sensor validation will address each of the different physical types of sensors that may be represented among the prospective measurements to be used by the expert system, including:

- Temperature
    - thermocouple
    - RTD
- Pressure
- Flow
    - orifice
    - flow nozzle
- Oxygen analyzer
- Carbon dioxide analyzer

- Ammeter
- Level

Consideration is also being given to the use of typical precision and accuracy values for each physical type of sensor, which may be replaced by the user with any appropriate instrument-specific values, if available. Although the expert system is being designed to address each of these types of measurements, the availability of a complete suite of on-line instruments will not be required for its operation.

CONCLUSIONS

An on-line expert system is being developed by Sargent & Lundy for the Electric Power Research Institute. It is intended to enable plant operators to evaluate more of the available information in greater depth to identify and respond to degradations in heat rate more quickly than is currently possible. The anticipated benefits of this advanced capability include increased plant efficiency and reliability.

The development of this expert system includes several significant features. Host and Demonstration Utilities are an integral part of the development project team. A Project Application Specification is being used to structure the development process and will also facilitate continuity with related expert system development projects at EPRI. The expert system is being designed to be configurable for most utility plants without modifications to the expert system or to the rules contained in its knowledge base. A modular architecture is being used for the expert system to provide the optimal combination of flexibility and performance. A user interface is being provided that conveys most routine information graphically while using a standard "look and feel" for feature selection consistent with the EPRIGEMS ™ Product Specification. An "open architecture" is being developed for the exchange of information between the expert system and potential sources of on-line data and results. Sensor validation techniques are being employed that will enable this expert system to interpret plant measurements with significantly greater sensitivity and accuracy than systems that do not specifically address sensor validation and that employ other techniques, such as knowledge base "tuning," to account for measurement inaccuracies.

The features being incorporated in the "Heat Rate Degradation Expert System Advisor" are ultimately expected to provide a sound foundation for a network of specialized expert system applications that will play a significant role in the future of power plant automation.

REFERENCES

1. Heat Rate Improvement Guidelines for Existing Fossil Plants. Report CS-4554. Palo Alto, California: Electric Power Research Institute, May 1986.

2. Cain, D., E. Choi, S. M. Divakaruni, V. Longo, and T. Wilson, "EPRIGEMS™: Expert Systems for Technology Transfer." Presented at the Electric Power Research Institute Seminar on Expert Systems Applications for the Electric Power Industry, Lake Buena Vista, Florida, June 5 - 8, 1989.

3. Expert Systems Technology Assessment and R&D Plan for Fossil Plant Applications. EPRI RP2819-5. Palo Alto, California: Electric Power Research Institute, April 1989.

4. Valverde A., Jr., L. J.; S. M. Gehl, A. F. Armor, J. Scheibel, and S. M. Divakaruni. "Fossil Power Plant Applications of Expert Systems: An EPRI Perspective." Presented at the 51st American Power Conference, Chicago, Illinois, April 24 - 26, 1989.

5. Divakaruni, S. M., G. Kozlick, D. M. Sopocy, and B. Frogner. "Review of Fossil Plant R&D Project Prioritization for AI Applications." Presented at the Electric Power Research Institute Seminar on Expert Systems Applications for the Electric Power Industry, Lake Buena Vista, Florida, June 5 - 8, 1989.

6. Executive Summary: R&D Plans for Advanced Computer and Control Technologies. EPRI RP2819-5. Palo Alto, California: Electric Power Research Institute, November 1988.

7. Sopocy, D. M., A. R. Glazer, J. K. Rice, and O. Jonas. "Expert Systems to Improve Plant Operability." Proceedings of the American Power Conference, Volume 50. Chicago: Illinois Institute of Technology, April 1988.

8. Rice, J. K. "Validation of Monitoring Methods." Presented at the Electric Power Research Institute Conference on Cycle Chemistry in Fossil Plants, Seattle, Washington, August 30 - September 1, 1988.

9. Agogino, A. M., S. Srinivas, and K. M. Schneider. "Multiple Sensor Expert System for Diagnostic Reasoning, Monitoring and Control of Mechanical Systems." Mechanical Systems and Signal Processing, Volume 2, Number 1, 1988.

Radwaste Decision Support System (Functional Specification)

GEORGE WESTROM, E. R. KURRASCH, and R. E. CARLTON
Odetics, Inc.
Artificial Intelligence and Senior Engineering
1515 South Manchester Avenue
Anaheim, California 92802-2907, USA

J. N. VANCE
Vance Associates

It is generally recognized that decisions relative to the treatment, handling, transportation and disposal of low-level wastes produced in nuclear power plants involve a complex array of many inter-related elements or considerations. Complex decision processes can be aided through the use of computer-based expert systems which are based on the knowledge of experts and the inferencing of that knowledge to provide advice to an end-user.

To determine the feasibility of developing and applying an expert system in nuclear plant low level waste operations, a Functional Specification for a Radwaste Decision Support System (RDSS) was developed. All areas of radwaste management, from the point of waste generation to the disposition of the waste in the final disposal location were considered for inclusion within the scope of the RDSS. The Specification is to be used as a basis for assessing the feasibility, the scope, and the application of the RDSS. The feasibility of developing the RDSS was based on the availability of knowledge and information for the decision process. Decisions and logic structure for the radwaste management functions were investigated to assess the scope and feasibility of developing the RDSS. The equations and algorithms associated with the RDSS functions and logic structure were investigated as a further aid in assessing the feasibility. The availability of waste treatment and processing performance data was investigated to determine the adequacy of existing data to support the development of the RDSS. The RDSS Functional Specification addresses the software architecture as an organizational structure of the expert system knowledge and data bases under executive control. The potential benefits of using an expert system in radwaste management operations were evaluated.

This Functional Specification addresses the following topics.

- Functions of the RDSS

- Relationships and interfaces between the functions

- Development of the decisions and logic tree structures embodied in waste management

- Elements of the database and the characteristics required to support the decision-making process

- Specific User requirements for the RDSS

- Development of the User interface

- Basic software architecture

- Concepts for the RDSS usage including updating and maintenance

The computer hardware and software requirements for the RDSS were evaluated to the extent necessary to support this project effort.

PERFORMANCE REQUIREMENTS

The performance requirements for the RDSS are directed at establishing a User-friendly graphic interface with access to important and significant databases and information at all key steps in the use of the system. In addition, the RDSS will be easily updated to reflect changes in dynamic areas such as the regulatory environment, processing technologies and disposal practices. Specific User requirements are:

- The User interface shall use techniques to make the RDSS usable by a large segment of the Users.

- The knowledge and databases shall be updated on a regular basis under configuration control for the production software.

- Upon request, the User shall be provided explanations for the decision process.

- The RDSS shall be made available as an aid in the training of personnel in selected areas of interest.

- An easy-to-use graphics interface shall be developed to assist the user in running all parts of the program.

- To the extent practical, relevant documentation supporting regulatory compliance shall be provided by the system.

- The plant specific databases shall be updated on a regular basis.

- When alternative approaches to solving a problem are identified and evaluated, the evaluation results and explanations.

- The ability to query the data and knowledge bases for specific information and relationships shall be provided.

- The RDSS shall be capable of providing formatted reports.

- Initialization of the RDSS will be limited to plant specific information; all other data and knowledge shall be supplied and routinely updated with source references.

- Documentation and verification shall meet the applicable portions of 10CFR50, Appendix B and the implementing ANSI standards.

- On line..Help..shall be available to the User to guide him in the proper use of the RDSS.

- Where practical, data input shall be monitored for acceptable parameter ranges.

- Integrate with external databases as feasible.

SCOPE OF THE RDSS

The project direction required that virtually all low level waste management activities be considered for inclusion within the scope of the RDSS.

One consideration is that the more areas where advice and assistance are provided in the performance of waste management activities, the greater the benefit to the User. A broad scope will allow different applications or uses of the RDSS, such as: process optimization on existing treatment operations, planning exercises for a major decontamination outage, and operations personnel training.

One consideration is that expert system software can be developed to deal with a large number of databases and different logic structures. Likewise, many of the databases, equations and algorithms developed can be made applicable to several waste types with little additional effort.

The RDSS shall address the following four types of low level wastes produced in PWR and BWR nuclear generating plants.

- Aqueous Liquid Wastes
- Organic Liquid Wastes
- Wet Solid Wastes
- Dry Solid Wastes

The RDSS logic shall provide the capability to address the above four waste types for the following three conditions:

- Routine Conditions (routine wastes)
- Upset Conditions (problematical routine)
- Non-routine Conditions (one-of-a-kind)

The RDSS shall provide advice, information to assist in processing and disposal decisions, and instructional material to accomplish the suggested actions. The RDSS provides assistance in the form of advice and information to the User on processing, transporting and disposing of plant low-level wastes within plant operating limits, regulatory limits, and in a cost-effective manner. The plant operating limits are those identified in plant technical specifications and operating procedures. The regulatory limits are related to the discharge, transportation and disposal of low-level solid wastes in regulations such as:

- NRC Regulations - Appendix I, 10CFR50
- NRC Regulations - 10CFR20
- EPA Regulations - 40CFR261
- EPA Regulations - 40CFR263
- DOT Regulations - 49CFR173
- NRC Regulations - 10CFR61
- NRC Regulations - 10CFR72
- Burial Site License Agreements
- Burial Site Criteria

For routine wastes, the RDSS shall provide information and advice which focuses principally on the volume and cost aspects of available processing and disposal options.

For problematical routine wastes, the RDSS shall provide advice and information that focuses primarily on processing or treatment alternatives to adequately treat and dispose of problematical waste.

For non-routine or one-of-a-kind wastes, the RDSS shall address virtually every aspect of dealing with the waste; including advice, information and instructional material on:

- Release or recycle of the waste
- Treatment by existing plant equipment
- Temporary treatment equipment
- Waste classification
- Packaging
- Transportation
- Waste form modification
- Disposal
- Cost
- Volume reduction
- Radiation levels
- On-site storage

PLANT-SPECIFIC INITIALIZATION OF RDSS

To use the RDSS, it will be necessary to perform an initialization for plant-specific parameters, plant specific information and databases, and to provide the radwaste processing configurations. It is generally expected that the initialization would be performed only once for a given plant. RDSS will be configured so as to interface with a specific version of the external computer codes such as LADTAP as listed in Table I. If a plant uses a different version of LADTAP or other radwaste codes, an interface module will be provided in the initialization process so as to make the parameters compatible with RDSS.

RADWASTE DECISION STRUCTURE

The logic and decision trees, which an individual performing the functions of an expert would follow in processing low level waste, have been developed as a function of waste types typical of both PWR and BWR plants. The decision trees for the four waste types are Organic Liquids, Aqueous Liquids, Dry Solids and Wet Solids. They are organized in a series of logical steps that are typically followed by a radwaste expert in a manual system. In order to make the necessary decisions, data is required, regulations must be referred to, computations are made, and frequently judgments and trade offs must be made.

Table II is a summary of the data requirements and external computer code usage for the four waste types. Table III and I provide a description of the types of data and the external computer codes respectively. It can be seen from the tables that many of the data requirements are common to many parts of the RDSS.

RDSS SOFTWARE ARCHITECTURE

The major components in the RDSS are as follows.

Inference Mechanism

The inferencing mechanism for Advisor will be selected from the available commercial expert system shells which operate in a PC environment. The selection of the shell will be based on the following criteria: Inference mechanism, Knowledge Representation scheme, Capability for front-end graphics, Limit on Number of Rules, Memory Storage Requirements, Language, Computer Host Hardware Requirements, Graphics Interface Capabilities.

EXTERNAL COMPUTER CODE DESCRIPTIONS

TABLE I

CC1	LADTAP or other Liquid Release Environmental Dose Computer Code
CC2	Radiation Transport (Shielding) Analysis Computer Code
CC3	Solid Radwaste Shipping Computer Code
CC4	Gaseous Effluent Computer Code

SUMMARY OF DECISION TREE REQUIREMENTS WASTE STREAM

TABLE II

Aqueous Liquid Waste	Organic Liquid Waste	Dry Solid Waste	Wet Solid Waste
DB1	DB4	DB17,CC3	DB14
	DB5	DB14	DB15
DB2,CC1	DB14	DB14	DB11
DB2	DB7	DB14	DB12,CC3
DB3	DB19	DB20	DB13
DB4	DB11	DB12,CC3	DB4,DB5
DB5	DB14	DB12,CC3	DB16,CC2
DB6		DB13	DB17,CC3
CC1	CC4	DB4,DB5	DB11,DB15
DB7	DB14	DB16,CC2	DB17,CC3
DB8	DB13	DB17,CC3	DB18
DB8	DB18	DB20	
	DB7	DB16,CC2	
DB8	DB12	DB18	
DB9	DB13		
DB8	DB17,CC3		
DB10	DB8,CC2		
DB8,CC2			
DB11			
DB12,CC3			
DB5			
DB13			
DB14			
DB15			
DB16,CC2			
DB17,CC3			
DB17,CC3			
DB18			

RDSS DATA REQUIREMENTS

TABLE III

DB0	Radionuclide Baseline Data
DB1	Plant Recycle Water Chemistry Parameters
DB2	Plant Technical Specifications for Discharge
DB3	10CFR20, Appendix B, Table II, Radionuclide Concentration
DB4	40CFR261, EPA Hazardous Chemical Concentrations
DB5	EP Toxicity Limits from 40CFR263
DB6	NPDES/State/Local Chemical Release Limits
DB7	Radionuclide Decontamination Factors for Process Equipment
DB8	Plant Process Equipment Performance and Geometry Factors
DB9	Temporary Equipment Performance Factors
DB10	Plant Radwaste System Construction Materials
DB11	Solidification System Performance Factors
DB12	10CFR61, Tables 1 and 2
DB13	Burial Site Disposal Restrictions
DB14	BRC Exemptions (10CFR20-302)
DB15	Dewatered Solid Radwaste Characteristics
DB16	Shipping Liners and Cask Parameters
DB17	Shipping Regulations (49CFR173)
DB18	Cost Data
DB19	Materials Excluded from Plant Boiler
DB20	Volume Reduction Performance Parameters

Knowledge Bases

The knowledge base contains the vital information and rules for this knowledge domain of radwaste handling and will represent all the knowledge, rules, and procedures required by the logic flow/decision tree diagrams. The important characteristics of the knowledge base are its organization, interface with the inference mechanism, interface with the databases, knowledge representation of the rules, and communication with the User interface.

DataBases

The global data facts and information for the RDSS are contained in a database organized such that specific information used only by any one of the four knowledge bases is linked associatively with that knowledge base.

Executive Controller

The RDSS Executive shall perform these features: User Input, Graphics/User Interface, Output, Menus, Explanation Facility, Display Mechanism (Advice, Information), User Functions (Edit, Update, Display, etc), Process Selection, User Help Function (Instructional Guidance), Interface Modules for Initialization. The executive controller shall be the main interface and link between the User and the RDSS expert system.

User Functions

The User functions for the RDSS are the following; Operate, Initialize, Input, Train, Help, Update, Maintain, Exit.

RDSS MAINTENANCE

As a major software logic system, the Radwaste Decision Support System (RDSS) will require a comprehensive set of maintenance activities to assure its optimum functioning. The scope of the system will include data bases (some of which will be periodically changing), knowledge bases which will change and improve with experience, and radwaste computer codes which may need correction or modification. The RDSS will require user training, facility specific modifications and interfacing to other utility-specific computer systems.

Therefore, it is recommended that a Central Maintenance Facility be established to maintain configuration control so that the software will provide all of the potential benefits.

APPLICATIONS AND BENEFITS

It is envisioned that many applications or uses of the RDSS would be identified by users in the implementation of such a program. However, in the development of the functional specification several applications have been identified as a basis for the structure and content of the RDSS. To provide a better understanding of how the RDSS might be used, the following presents a summary description of some of the more important applications. The identified applications are as follows:

- A planning tool which gives the plant a rapid capability to accurately assess the total impact of the creation of a non-routine type of waste before its creation.

- A historical record data base which gives the end-user the benefit of the major experiences other users have had in radwaste management.

- Assist in the optimization of the performance of the existing plant system with respect to operating costs and waste volumes generated.

- Providing expertise in the handling of radwaste problems that occur on an infrequent basis but require knowledge and expertise beyond routine instructions.

- Identifying means to overcome the undesirable effects of malfunctioning equipment in the radwaste system or other key interfacing systems in the plant.

- A primary training tool for the radwaste system operators and engineering support staff.

- An engineering evaluation tool to assist in studying alternative processing methods or alternative volume reduction techniques.

- A regulatory compliance aid to backup and provide a check against existing compliance procedures.

Development of Isolation Support System for Nuclear Power Plants

TATSUO YOSHIKAWA
Nippon Atomic Industry Group Co, Ltd
System Analysis Department
4-1 Ukishima-cho, Kawasaki-ku
Kawasaki, 210 Japan

TAMAYOSHI OBARA and DAIJIRO IKEDA
Toshiba Corporation
Nuclear Control and Electric Engineering Department
8 Shinsugita-cho, Isogo-ku
Yokohama, 235 Japan

SHINJI IWATA
Chubu Electric Power Co., Inc.
Nuclear Power Operation Department
1 Toshin-cho, Higashi-ku
Nagoya, 461 Japan

ABSTRACT

When components of a power plant such as pumps or valves are checked or repaired, these components should be isolated. This isolation work follows the work sheet which describes components to be operated. This work sheet is called the isolation list and prepared by technical experts. Technical experts have to refer to many drawings to prepare the isolation list and this task is a rather time consuming and complicated. Isolation Support System has been developed to support an engineer while preparing an isolation list. The frame based representation of components and the production rule based representation of the isolation procedure are used to develop the isolation operation which establish the isolation boundary and drainage. The switching interlocks are specified by simulating the plant response to the isolation operation. Technical experts tested Isolation Support System using a reactor water cleanup system model and the validity of this system was confirmed.

INTRODUCTION

When components of a power plant are maintained during the plant operation or the routine maintenance, the isolation work is performed to keep safety for the workers and not to disturb the operating plant systems. Isolation work includes equipments stop, power supplies cut, valves close, valves open and so on. This isolation work follows the isolation list which is a work sheet and describes title and period of the work and components to be operated. Technical experts prepare the isolation list tracing piping and electrical lines across many drawings such as Piping and Instrumentation Diagrams (P&IDs) and Elementary Wiring Diagrams (EWDs) considering personnel and equipment safety. This preparation task is a rather time consuming and complicated. There have been systems which assist preparing the isolation list by searching database which contains many isolation lists previously developed manually. Preparing the

isolation list consistent with the plant condition using these systems, there is necessity to modify the list referring to the many drawings of the plant (P&IDs,EWDs). On the other hand, systems applying Artificial Intelligence (AI) technology have been popular in the industrial, and plant diagnosis systems and so on have been developed in the electric power industry. In these industrial background, to ascertain the validity of AI application to the isolation work, Isolation Support System has been developed as a model system which supports an engineer while preparing the isolation list within mechanical components of the reactor water cleanup system (CUW). The frame based representation of the plant system components and the production rule based representation of the isolation procedure are processed to develop the isolation list. Technical experts tested the model system and the validity such as proper routes selection for the drainage, proper boundary selection and prevention of failure to notice switching interlocks to the isolation was confirmed. This paper describes how to develop an isolation list applying AI technology and examples of the system examination.

SYSTEM CONFIGURATION AND AN OVERVIEW OF THE PROCESSING

Figure 1 shows the configuration of Isolation Support System. This system takes title and period of the maintenance work, components to be maintained and type of the isolation (individual or lump). The system processing flow is as follows :

(1) search routes for the specified components drainage
(2) search boundary valves for the specified components
(3) infer isolation operation applying rules for the isolation procedure to the boundary valves, components of the bounds, the routes for drainage and the components to be maintained
(4) simulate the plant system status after the inferred isolation operation then predict switching interlocks and annunciators.
(5) infer isolation operation again considering the predicted switching interlocks and annunciators.
(6) check consistency between the developed isolation operation and the saved isolation operations which overlap period with the developed.
(7) If there is inconsistency then the system assists preparing the other isolation list which is consistent.

The plant system constitution data, rules for the isolation procedure and saved isolation lists are used in each processing block. The system displays components to be operated for the isolation in pre-defined color which is able to identified operation meaning by a user on the P&ID window of the CRT. It also displays and prints out the isolation list. Figure 2 shows an example of the system CRT display. The lower left window is the system control window through which a user input title and period of the maintenance and type of the isolation. Identification of the components to be maintained is accomplished by clicking a mouse button on the corresponding symbols of the P&ID window. Isolation list and scheduling table are also displayed on the CRT as shown in figure 2.

KNOWLEDGE REPRESENTATION

The Plant System Constitution Data

The plant system constitution data is classified and hierarchically structured so that properties of lower classes inherit from upper classes as shown in figure 3 and represented in frames. The components of the plant system are not

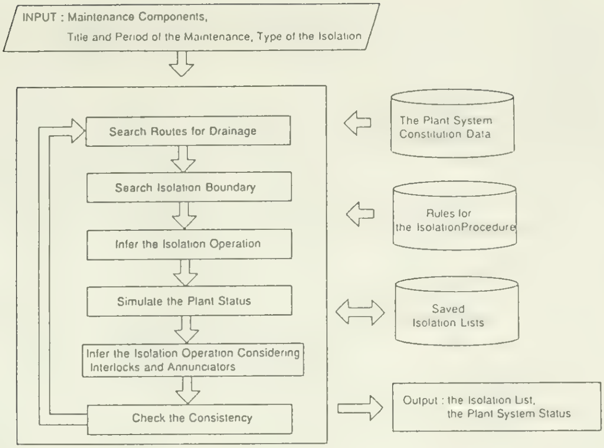


Figure 1. Configuration of Isolation Support System

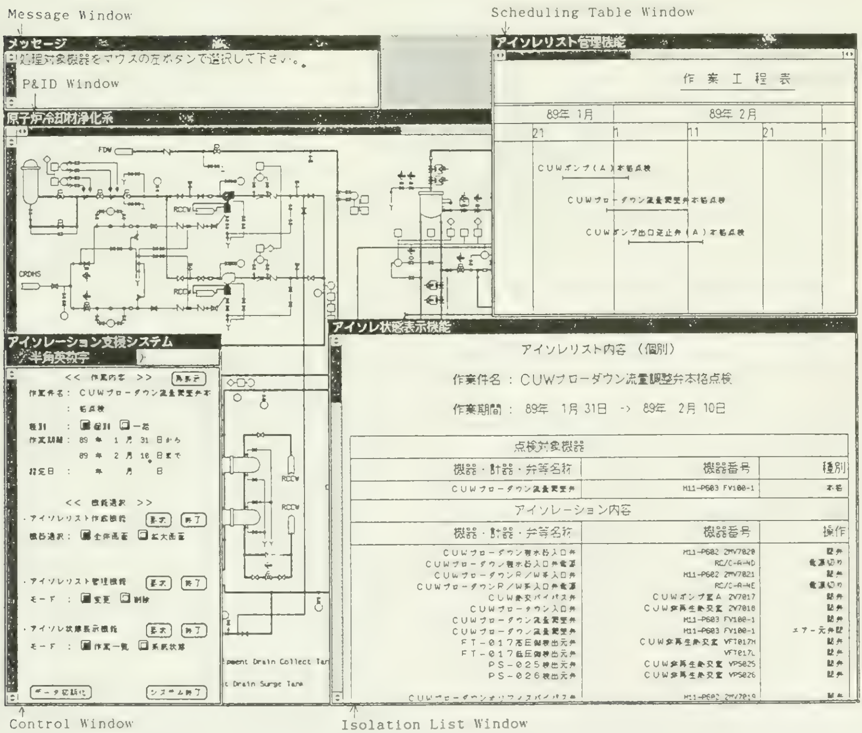


Figure 2. an Example of the System Display

shown in figure 3, but they exist in lower classes than those in figure 3. To represent connection and elevation relation of the plant components, the component which has multiple inlet or outlet is represented by a set of elementary components and nodes, the former have only one inlet and one outlet, the latter have multiple inlet or outlet but have no other spatial properties. Principal properties of an elementary component are name of component which connect to inlet and outlet, length and elevation between inlet and outlet and inside diameter. These data are extracted referring to P&IDs, piping isometric drawings and equipment structure drawings. There are 2534 frames which include 1133 equipments, 1054 pipes and 289 nodes for the reactor water cleanup system of Chubu Electric Power Company Hamaoka Nuclear Generating Station Unit 2 (H-2).

Rules for the Isolation Procedure

Rules for the isolation procedure were extracted asking H-2 plant staffs how they prepare isolation lists for mechanical components of the reactor water cleanup system. The rules were classified as follows :

(a) Rules which determine the isolation boundary
(b) Rules which determine the isolation operation using boundary, components of bounds, routes for drainage and components to be maintained
(c) Rules which determine the isolation operation using interlocks and annunciators related to the operation

Table 1 shows examples of rules for the isolation procedure. Above class (a) rules are isolation operation determinative rules because when the isolation boundary is determined with those rules, the isolation operation is fixed with above class (b) and (c) rules without exception. All of class (a) rules are followings :

(1) Route to drain fluid away from the maintenance components should be exist
(2) Lesser volume of the bounds case is better
(3) Drainage fluid into the plant system main tanks (ex. main condenser) case is better than that into the drain funnels case
(4) Lesser number of operation valves case is better
(5) Motor or air operation valves are better than manual operation valves to be operated
(6) Lesser radioactivity valves are better to be operated
(7) Lesser influence to the out of the bounds case is better

To get the best boundary which satisfies these rules, the function which estimates degree of pertinent is necessary. The function, for example, estimates which is better 60 liter volume of bounds 10 manual operation valves 1 motor operation valve case or 80 liter volume of bounds 8 manual operation valves 3 motor operation valves case. Such a function is difficult to make because of obscurity, so that following strategy was used to determine the boundary.

(1) search routes for drainage aiming at the plant system main tanks
(2) search routes for drainage giving priority to lesser volume of the bounds when the plant system main tanks could not be found
(3) search boundary so that routes for drainage exist in the bounds
(4) search next prior routes for drainage or change boundary or add boundary according as a user's decide

The method to find routes for drainage and that to find boundary are mentioned later. The production rules (if-then-rules) based representation was used for the above class (b) and (c) rules. There are 45 common rules which are able to be used for the other plant systems also and 11 peculiar rules which are able to be used for the reactor water cleanup system only.

Table 1.

CLASS AND EXAMPLES OF RULES FOR THE ISOLATION PROCEDURE

CLASS	EXAMPLES OF RULES FOR THE ISOLATION PROCEDURE
(a) Rules which determine the isolation boundary	Route to drain fluid away from the maintenance components should be exist
	Lesser volume of the bounds case is better
(b) Rules which determine the isolation operation (not consider interlocks nor annunciators)	If a component is a boundary and a motor operation valve then the power supply should be cut
	If a component is in the bounds and a pump then it should be stopped
(c) Rules which determine the isolation operation (consider interlocks and annunciators)	If a component stops with the interlock then it should be stopped
	If a component is a valve and closes with then interlock then it should be closed

SYSTEM PROCESSING

The Method to find Routes for Drainage and Boundary

When components of a plant piping system are maintained, fluid (water mostly) in the components should be drained away. The drainage is performed establishing the boundary (closing valves which surround the components to be maintained) and opening valves which are on the drain-able routes, so that it is necessary to find routes for drainage. The routes for drainage consist of routes for fluid outpouring (called drain routes) and routes for air intake (called vent routes). Drain routes are found tracing piping from the components to be maintained to the plant system main tanks or the drain funnels unless piping is higher than the components to be maintained. Vent routes are found tracing piping from the components to be maintained to the drain funnels unless piping goes down and then goes up. Figure 4 shows an example of a drain route and a vent route. In this tracing process, connection and elevation data represented in frames are used. Boundary valves are found tracing piping so that routes for drainage exist in the bounds. The type of the valve is processed to determine if it is used as a boundary valve. For example an air operation valve which is open in case of failure of the air supply is not used as a boundary valve.

The Method to Simulate the Plant Status

The plant response to the isolation operation should be predicted to consider the switching interlocks and annunciators into the isolation operation. Assuming that flow is directly proportional to pressure difference and inversely proportional to flow resistance, equations including flow and pressure variables are constructed tracing connections of the plant components. Solving the equations, values of flow and pressure variables are obtained. Assuming specific heat of fluid is constant, equations of heat balance are constructed tracing connections of the plant components. Solving the equations, values of fluid temperature variables are obtained. Each component of the top stream has a pressure and a temperature property and these values should be set previously. Flow resistance of each component is calculated with it's inside diameter, length and ratio of choked area to piping area using brief equation which is independent to type of the component. Each pump has a pump head property and it's value previously set. Head of each component is calculated with elevation between the inlet and the outlet. Each component which generates heat has a heat generation ratio property and it's value. Each heat exchanger has a heat transfer ratio property and it"s value. Assuming flow is directly proportional to pressure difference and inversely proportional to flow resistance, piping system is transformed to electrical system as shown in figure 5. Tracing connections of components from the top stream, voltage variables (\$E1,\$E2) are set in nodes and electric current variables (\$I1,\$I2,\$I3,\$I4) are set in resistances and equations are constructed. Equations are as follows :

$$E0+E1+E2-(R1+R2)X\$I1=\$E1 \quad .....(1)$$
$$\$I1=\$I2+\$I4 \quad .....(2)$$
$$\$E1+E3-(R3+R5+R7+R9+R11+R13)X\$I2=\$E2 \quad .....(3)$$
$$\$I2+\$I4=\$I3 \quad .....(4)$$
$$\$E2+E5-R5X\$I3=E0 \quad .....(5)$$
$$\$E1+E4-(R4+R6+R8+R10+R12+R14)X\$I4=\$E2 \quad .....(6)$$

(E0...E5,R1...R14 are constant in these equations)

Solving these equations, the value of each variable is obtained. Voltage is translated to pressure and electric current is translated to flow. Stopping a pump is simulated by setting the corresponding voltage to zero. Closing a valve is simulated by setting the corresponding current to zero and eliminating the

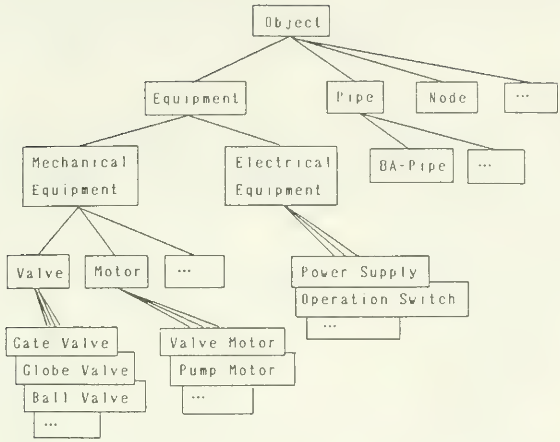


Figure 3. Hierarchy of the Plant Constitution Data

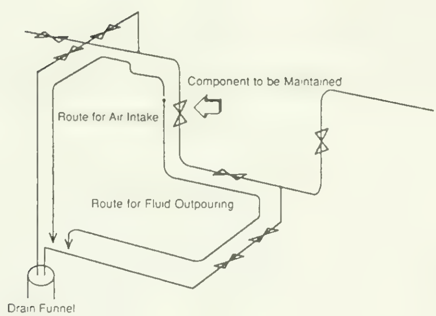


Figure 4. Routes for Drainage

kirhihof first law equation which includes the current variable. If there is reverse flow in a check valve then calculate again closing the check valve.

Inference_Engine

Equipments to be stopped, power supplies to be cut, valves to be closed or opened, positions of electrical lines to be lift or jumpered, equipments to be keylocked and attention items are obtained applying IF-THEN rules for the isolation procedure to the boundary valves, components of the bounds, components on the routes for drainage, the switching interlocks and annunciators.

Consistency_Check

Checking consistency between the developed isolation operation and the saved isolation operation which overlap period with the developed, if there is inconsistency then the system assists preparing consistent isolation operation. Users are able to select one of followings to prepare consistent isolation operation.

(1) Change boundary finding the other routes for the drainage
(2) Spread boundary which includes bounds of the inconsistent
(3) Change period of the maintenance work

EXAMPLES OF THE SYSTEM EXAMINATION

Figure 6 shows the P&ID window of the reactor water cleanup system. When the plant system is in normal operation, reactor water is pressured by two parallel CUW pumps, cooled by five serial heat exchangers, decontaminated by two parallel filter demineralizers, heated by two heat exchangers and return to the reactor pressure vessel through the feed water system. Preparing isolation lists for 40 maintenance works, the validity of the system was confirmed as follows :

(1) Sure routes for drainage and boundary are obtained utilizing the plant system constitution data which includes component connection and elevation relation
(2) Able to prepare switching interlocks and annunciators considered isolation lists utilizing the plant system simulation
(3) Able to prepare consistent isolation list checking the developed isolation operation consistency with the other isolation operations
(4) Able to prepare an isolation list 1/10 of time comparing with the technical experts preparing time without computer support

Figure 7 shows an example of the system display when the CUW blow down flow control valve is selected for the maintenance. In this figure the boundary, the bound and the route for drainage are shown. In this case an outpouring route is identical to an air intake route. If a user who is not familiar to the plant system examines the isolation operation referencing to P&ID, he may try to use the other routes for outpouring. The system shows that a route shown in figure 7 is the only one route for outpouring. Figure 8 shows an isometric piping drawing around the specified valve and it is apparent that the system shown route is the only route for outpouring. Figure 9 shows an another example of the system display when the CUW pump inlet first valve (A) is selected for the maintenance. The route via 2V7006A, 2V7802AX and 2V7802AY is obtained as an outpouring route and the route via 2V7027X and 2V7027Y is obtained as an air intake route. 2MV7003, 2MV7004, 2V7008A, 2V7017 and 2V7049A are obtained as the boundary valves. Simulating the plant status after inferred isolation operation, interlocks predicted to be switched are followings :

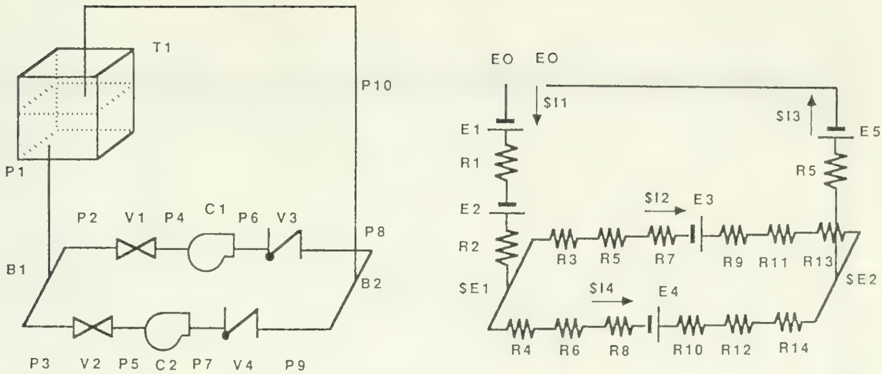


Figure 5. Analogy between Piping System and Electrical Circuit

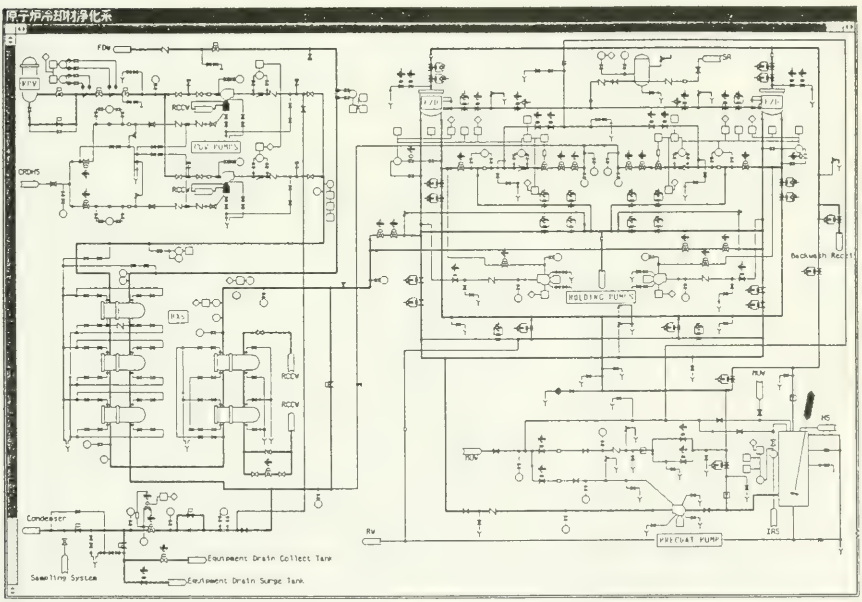


Figure 6. the P&ID Window of the Reactor Water Cleanup System

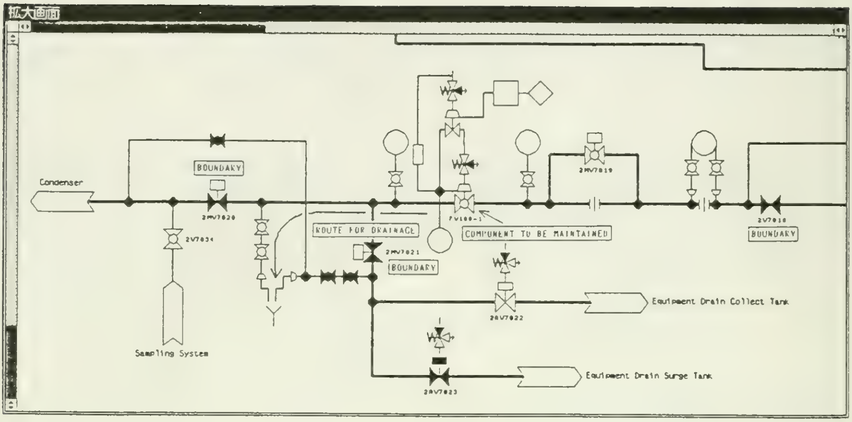


Figure 7. an example of the system display that shows an route for drainage and boundary valves.

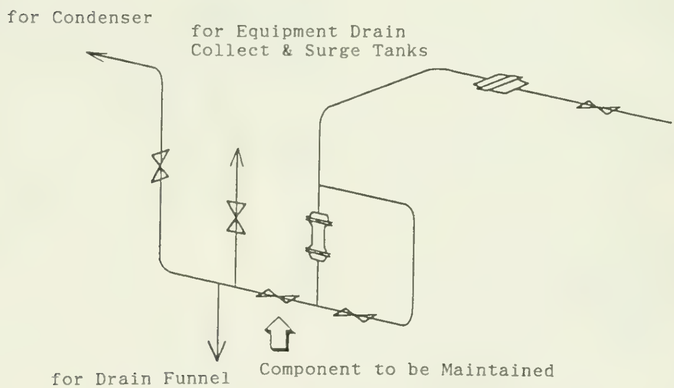


Figure 8. the Isometric Piping Drawing around the CUW Blow Down Flow Control Valve

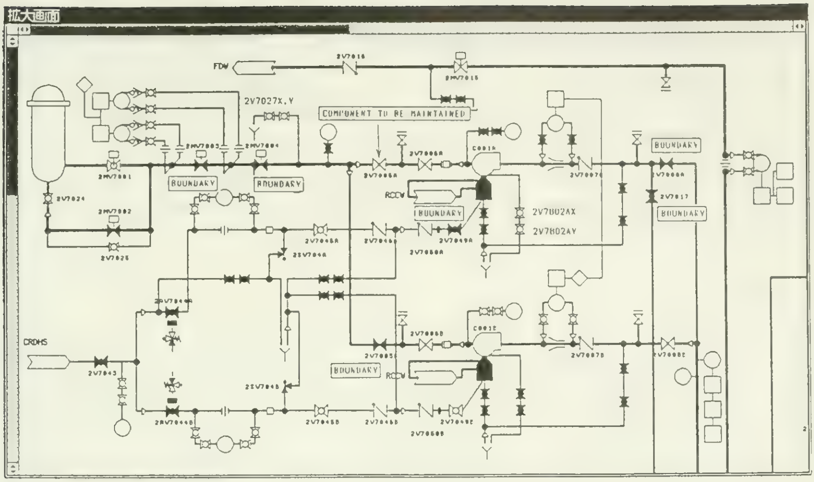


Figure 9. an example of the system display that shows isolation operations on the P&ID window.

アイソレリスト内容 (個別) <-- Isolation List (Individual)

作業件名 : ＣＵＷポンプ入口第１弁（Ａ）本格点検 <-- Title of the Work : CUW Pump Inlet First Valve (A) Check
作業期間 : 88年 12月 27日 -> 88年 12月 31日 <-- Period of the Work : Dec 27,1989 - Dec 31,1989

点検対象機器 Component(s) to be maintained			
Name 機器・計器・弁等名称	Number 機器番号	Type 種別	
ＣＵＷポンプ入口第１弁（Ａ）	ＣＵＷポンプ室Ａ 2V7005A	本格	
アイソレーション内容 Isolation Operation(s)			
Name 機器・計器・弁等名称	Number 機器番号	Operation 操作	
ＣＵＷポンプ（Ａ）	C001A	停止	<-- stop
ＣＵＷポンプ（Ａ）電源	RC/C-A-2F	電源切り	<-- power off
ＣＵＷポンプ（Ｂ）	C001B	停止	
ＣＵＷポンプ（Ｂ）電源	RC/C-B-2F	電源切り	
ＣＵＷパージライン止メ弁（Ａ）	H11-P602 2AV7044A	閉弁	<-- close
ＣＵＷパージライン止メ弁（Ａ）	H11-P602 2AV7044A	エアー元弁閉	<-- close the air stop valve
ＣＵＷパージライン止メ弁（Ａ）電磁弁	H11-P602 T7-19ク T7-23キ RL-14	リフト	<-- lift the electrical line
ＣＵＷパージライン止メ弁（Ｂ）	H11-P602 2AV7044B	閉弁	
ＣＵＷパージライン止メ弁（Ｂ）	H11-P602 2AV7044B	エアー元弁閉	
ＣＵＷパージライン止メ弁（Ｂ）電磁弁	H11-P602 T7-26ク T7-30キ RL-14	リフト	
ＣＵＷポンプ入口第１隔離弁	H11-P602 2MV7003	閉弁	
ＣＵＷポンプ入口第１隔離弁電源	ERC/C-A-10B	電源切り	
ＣＵＷポンプ入口第２隔離弁	H11-P602 2MV7004	閉弁	
ＣＵＷポンプ入口第２隔離弁電源	DCC/C-B-12B	電源切り	
ＣＵＷポンプ入口第１弁（Ｂ）	ＣＵＷポンプ室Ｂ 2V7005B	閉弁	
ＣＵＷポンプ出口弁（Ａ）	ＣＵＷポンプ室Ａ 2V7008A	閉弁	
ＣＵＷ熱交バイパス弁	ＣＵＷポンプ室Ａ 2V7017	閉弁	
ＣＵＷパージライン元弁	ＣＵＷポンプ室前 2V7043	閉弁	
ＣＵＷパージラインポンプ元弁（Ａ）	ＣＵＷポンプ室Ａ 2V7049A	閉弁	
ＦＳ－０１３（Ａ）高圧側検出元弁	ＣＵＷポンプ室Ａ VFS013AH	閉弁	
ＦＳ－０１３（Ａ）低圧側検出元弁	ＣＵＷポンプ室Ａ VFS013AL	閉弁	
ＰＩ－０１９検出元弁	R/B 2FL 配管バルブスペース室 VPI019	閉弁	
ＰＸ－０２９（Ａ）試験圧力検出第１元弁	ＣＵＷポンプ室Ａ VPX029AX	閉弁	
ＰＸ－０２９（Ａ）試験圧力検出第２元弁	ＣＵＷポンプ室Ａ VPX029AY	閉弁	
ＣＵＷポンプ入口第１弁（Ａ）	ＣＵＷポンプ室Ａ 2V7005A	開弁	<-- open
ＣＵＷポンプ入口第２弁（Ａ）	ＣＵＷポンプ室Ａ 2V7006A	開弁	
ＣＵＷポンプ入口第１ベント弁	R/B 2FL 配管バルブスペース室 2V7027X	開弁	
ＣＵＷポンプ入口第２ベント弁	R/B 2FL 配管バルブスペース室 2V7027Y	開弁	
ＣＵＷポンプ（Ａ）第１ベント弁	ＣＵＷポンプ室Ａ 2V7802AX	開弁	
ＣＵＷポンプ（Ａ）第２ベント弁	ＣＵＷポンプ室Ａ 2V7802AY	開弁	
		Ｆ／Ｄ（Ａ）待機または停止	<- standby or stop
		Ｆ／Ｄ（Ｂ）待機または停止	

Figure 10. an example of the system display that shows an isolation list for the CUW pump inlet first valve (A) maintenance.

(1) C001B (CUW pump B) tripped because 2MV7003 or 2MV7004 is not full opening
(2) 2AV7044A (CUW pump A purge line stop valve) closed because C001A stopped
(3) 2AV7044B (CUW pump B purge line stop valve) closed because C001B stopped
(4) CUW holding pump A (shown in figure 6) worked because flow of the outlet of the filter demineralizer A is low
(5) CUW holding pump B (shown in figure 6) worked because flow of the outlet of the filter demineralizer B is low

Figure 10 shows the isolation list. In addition to operation for the boundary and components of the bound, operation for the components which are switched by the interlocks are listed.

CONCLUSION

Isolation Support System has been developed as a model system which supports an engineer while preparing the isolation list within mechanical components of the reactor water cleanup system. Technical experts tested the model system and the validity such as proper routes selection for the drainage, proper boundary selection and prevention of failure to notice switching interlocks to the isolation was confirmed. A study on Isolation Support System for practical use is under development. This system is planned to be used practically in 1992.

FOSSIL POWER PLANT APPLICATIONS

Review of Fossil Plant R&D Project Prioritization for AI Applications

S. MURTHY DIVAKARUNI
Electric Power Research Institute
3412 Hillview Avenue, Palo Alto, California 94304, USA

GREG KOZLIK
Northern Indiana Public Service Company

DALE M. SOPOCY
Sargent & Lundy

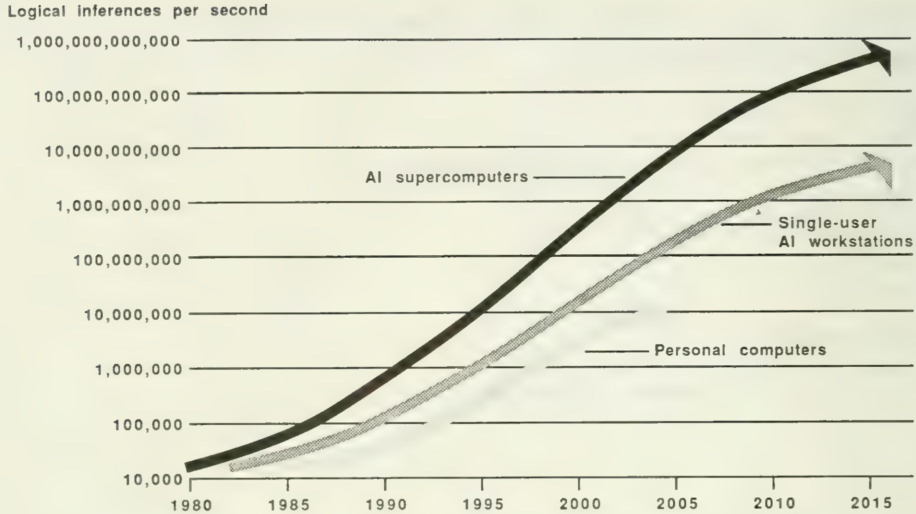
BJORN FROGNER
Expert-EASE Systems, Incorporated
1301 Shoreway Road, Belmont, California 94002, USA

ABSTRACT

This paper summarizes an R&D plan for an organized program of expert system application development that is designed to advance the implementation of expert systems technology throughout the utility industry. This plan has been developed through the efforts of a formal working group comprised of key representatives of utilities, manufacturers, architect/engineers, and consultants having experience in expert systems development. The results of a screening process by which the best prospective expert system applications were identified are presented in the form of summaries that address the need, advantages, limitations, research methodology, estimated costs and benefits, and commercialization potential of each application.

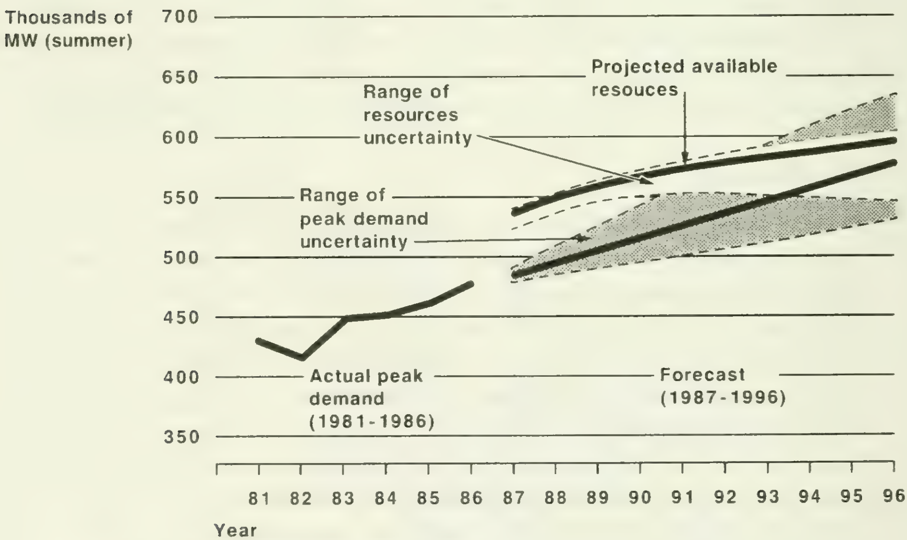
INTRODUCTION

The rapid growth of microprocessor technology, coupled with significant advances in the development of software shells, have caused the knowledge processing power of expert systems to grow exponentially over the past decade. This growth is projected to continue for the next two to three decades, as illustrated in Figure 1 (1). As a result, a wide variety of expert system applications are being developed for practical use in fields as diverse as medicine, finance, aerospace, manufacturing, military, and chemical process. Although there is a high level of interest in expert systems technology in the electric utility industry, implementation of expert systems in the utility industry is lagging behind other industries. A number of practical considerations could lead to a significant acceleration in the rate of expert systems implementation in the utility industry within the next several years, however, if the needed stimulus, a string of several successful expert system implementations, is provided.



Source: PC Week, October 31, 1988. M2341.003 11-88

Figure 1—The Growing Processing Power of Expert Systems Software



Source: National Electric Reliability Council

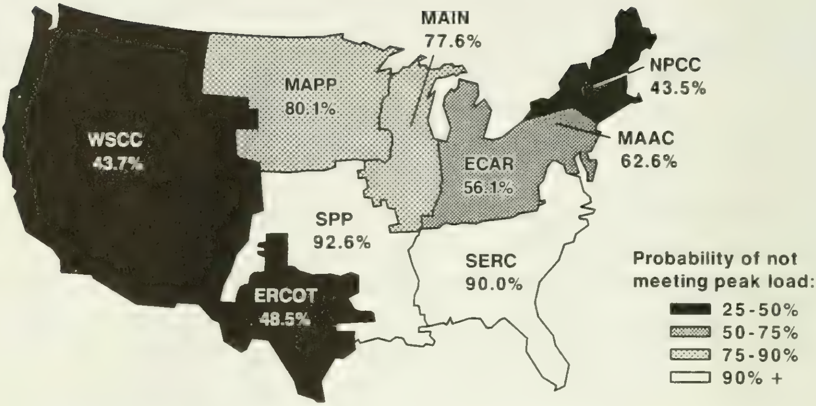
Figure 2—Projected Peak Demand and Available Resources

One of these considerations is declining reserve margins. In spite of the measures expected to be taken to extend the life and improve the performance of existing fossil fuel power plants, the available reserve margin of U. S. generating capacity is projected to decrease steadily over the next decade, as illustrated by Figure 2 (2). One potential impact of this situation is an increasing risk of electricity supply interruptions, as illustrated in Figure 3 (3).

Ultimately, many utilities are going to be faced with the choice of whether to handle a projected shortfall in generating capacity by adding gas turbines, encouraging additional cogeneration, or building new fossil fuel power plants. Those that elect to build new fossil fuel power plants will want designs that exceed the goals currently being sought for existing plants in heat rate, availability, and maneuverability, while having a low construction cost. There is a limited window of opportunity for the development of advanced technologies that will satisfy these needs in time to be incorporated in this next generation of fossil fuel power plants. Given the time of the projected shortfall in generating capacity and the lead time for plant design and construction, these technologies will need to be fully developed and proven by the mid-1990s if they are to be sufficiently assimilated across the utility industry by the late 1990s for confident use in the next generation of fossil fuel power plants. Furthermore, prudent planning and incremental, truly integrated development of the components of these technologies is needed so that shorter-term benefits may be realized through their incorporation in the refurbishment/improvement of existing units that will continue to take place throughout this period.

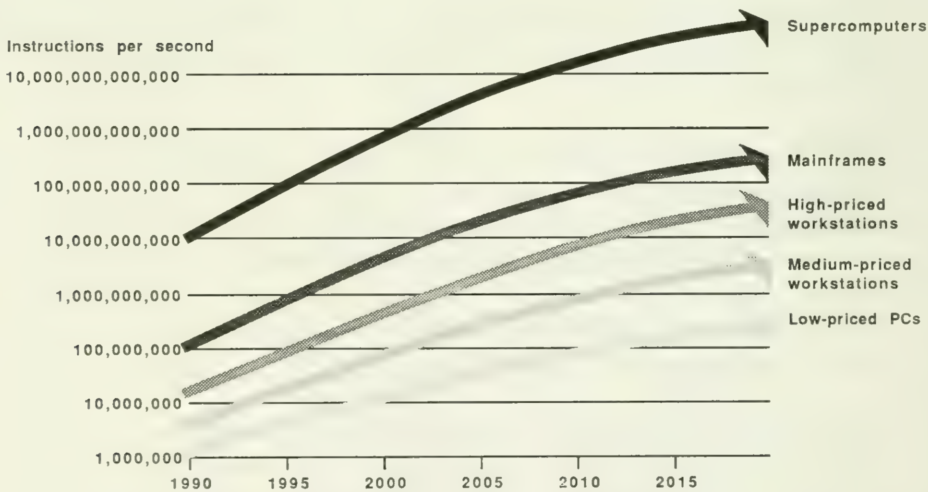
The needs for advanced technologies to support current plant life extension requirements and future new plant construction are self-evident. There are several additional factors, however, that make the development of advanced computer and control technologies, such as expert systems, virtually imperative for the power industry. These include demands for improvements in plant operability, reliability, and performance, as well as the issue of information management, which has an indirect impact on each of the other factors.

Power plant controls and instrumentation vendors now offer microprocessor-based controls, CRT-based control rooms, and a host of other automation systems for power plant operation and management. Coincidentally, operator culture is changing, requiring more advanced controls. The incorporation of expert systems will further



Source: "The Adequacy of U.S. Electricity Supply Through the Year 2000," AER Co., May 1987.

Figure 3—Forecast Risk of Electricity Shortages in the Year 2000



Source: PC Week, October 24, 1988.

M2341.004 11-88

Figure 4—The Growing Processing Speed of Computer Hardware

enhance power plant operation by enabling higher productivity, increased flexibility, and better performance than can be achieved by improved instrumentation and controls or equipment modifications alone.

While the automation and computerization of a variety of functions and tasks within fossil power plants have been accomplished by equipment manufacturers and computer system and control vendors, the more basic issue of integration has been ignored. This has created "islands of automation" and "islands of data." The various computer systems that are available for different functions throughout the plant are completely independent, with an inability to communicate with each other and a lack of a uniform user interface, and act as "islands of automation." There is also a lack of common databases which can be shared between the systems. Instead, each computer system typically has a separate database, constituting "islands of data." The continued implementation of independent systems is creating a situation where plant operators will be receiving too much simultaneous information to fully assimilate and interpret all of the available information. This situation, which would be most severe in new plants utilizing independent systems, is intolerable in that operators are being placed in a position to fail. Properly integrated, expert systems offer the potential of not only avoiding this situation, but actually increasing the success rate of operators in making the best plant operating decisions in terms of heat rate and unit life (4).

EPRI's CONTROLS AND AUTOMATION PROGRAM

EPRI has launched an initiative that will address these needs for both existing installed capacity and future power plant construction using recent advances in computer and control technologies, including expert systems. Computer technology has been advancing at an exponential rate and is expected to continue to evolve at this rate for the next three decades, as illustrated by Figure 4 (5). The EPRI initiative, the Controls and Automation Program, is intended to harness computer technologies to address the short-term needs of the power industry while building a foundation of advanced technologies that will benefit the industry in the long term.

The Controls and Automation Program is comprised of the following three integrated strategic program elements for the development of related advanced computer and control technologies:

- Controls and Instrumentation - Aimed at improving plant performance by designing more intelligent controls that incorporate inherent modeling of plant dynamics and process prediction capabilities, and by developing new instrumentation techniques.

- Expert Systems - Designed to capture available human expertise in various power plant technical disciplines, provide expert assistance to all levels of users, and train less experienced personnel.

- Automation and Simulators - Designed to fully exploit the power of computers to integrate and process information efficiently according to each of various end user needs.

The planning for the Controls and Automation Program is in place, with R&D plans for the three strategic program elements having recently been completed by independent formal working groups consisting of a total of 41 utility, 14 manufacturer, and 45 consultant representatives. These three R&D plans provide direction to the Controls and Automation Program and prescribe a total of 80 projects to be undertaken in 13 areas over the next 10 years. The total funding requirements for the three programs, exclusive of technology transfer, are estimated to be approximately $35 million over a 10-year period (4).

All of the expert system development projects in the Controls and Automation Program have been selected on the basis of having a relatively high benefit-to-cost ratio and short payback period based on tangible, direct benefits. Consequently, the potential benefits to be realized by this program are substantial. For example, the total potential benefits that have been estimated to be gained by those utilities projected to ultimately make use of the twenty expert systems that have been recommended for development range from $36,000,000 to $75,000,000 per year (6). This constitutes a significant return of investment considering that the total estimated funding requirement for the development of these expert systems is $7,000,000 (6). These projected benefits are based on conservative estimates of the benefits to be attained and the number of ultimate users.

This paper summarizes the expert systems R&D plan and provides an overview of the goals and benefits of the EPRI program of R&D projects that are intended to bring expert systems technology to the point of practical implementation in fossil fuel power plants across the utility industry. The approach used to identify, select, and prioritize expert system development projects in this R&D plan are well-suited for use by any organization considering development of an expert system for fossil fuel power plants.

EXPERT SYSTEMS R&D PLAN FOR FOSSIL FUEL POWER PLANTS

EPRI's R&D Plan for fossil fuel power plant expert systems development is detailed in Expert Systems Technology Assessment and R&D Plan for Fossil Plant Applications

(EPRI RP2819-5) (6). The principal utility investigators who contributed to the development of this report were

- G. Kozlik, Northern Indiana Public Service Company (Chairman),
- T. Anzolut, Consolidated Edison Company of New York, Incorporated,
- J. Davis, Duke Power Company,
- K. Guy, Public Service Company of Indiana, Incorporated,
- K. Legg, Southern Company Services, Incorporated,
- P. Mitchell, Niagara Mohawk Power Corporation, and
- A. Sudduth, Duke Power Company.

The EPRI Contractors for the preparation of this report were Expert-EASE Systems, Incorporated, and Sargent & Lundy.

Identification of Prospective Applications

The preceding EPRI conference, Expert Systems Applications in Power Plants, held in Boston, Massachusetts in May 1987, identified a number of expert systems that are in various stages of development and implementation for fossil units. Although a number of these systems were operational, the majority of the systems described at this conference were in the prototype stage of development at that time and represented first time efforts for the developers involved. A variety of prospective expert system applications was identified by working groups at this conference. Subsequent to the Boston conference, a utility interest group was formed by EPRI. The first meeting of the EPRI Formal Working Group on Expert Systems Technology, in September 1987, expanded the list of prospective expert system applications that was developed at the Boston conference. Each of these applications was categorized according to one of the following three major types of functionality: plant operations, equipment diagnostics, or information management. A total of fifty prospective expert system applications were ultimately identified. A variety of factors was considered in the identification of prospective expert system applications. Some of these factors, specific to the type of expert system functionality, are discussed as follows.

Plant Operations. Prospective expert system applications for plant operations were identified according to the following considerations:

- potential for improving equipment and/or unit performance and/or availability,

Table 7

APPLICATION RANKING

Application	Rating	Priority
Intelligent User Interface to Fuel Cost Model/Evaluator	7.3	Higher
Demonstrate Delivery of Expert System with EPRI Diagnostic Manual	6.7	Higher
Interface/Format for Distributed Info Processing Between ESs	6.4	Higher
Equipment Overhaul Planning Database/Advisor	6.1	Higher
Unit Executive Advisor	6.0	Higher
Boiler Tube Failure Inspection/Prediction	5.9	Higher
Cycle Chemistry Advisor	5.9	Higher
Turbine-Generator Diagnostics (Balancing)	5.8	Higher
Boiler Thermal Performance	5.6	Higher
Combustion Process Performance Analysis	5.6	Higher
Condenser/Feedwater Heater Diagnostics	5.6	Higher
Precipitator Diagnostic Trending	5.6	Higher
Repair Parts Management Database/Advisor	5.6	Higher
Unit Cycling Advisor	5.6	Higher
Demonstration of Live, Intelligent Database (Materials Experience)	5.5	Higher
Boiler Feed Pump and Driver Condition Diagnostics	5.4	Moderate
Predictive Maintenance - On-Line Failure Prevention	5.4	Moderate
Fan Inspector	5.3	Moderate
Painting and Coatings Advisor	5.3	Moderate
Wet FGD System Chemistry Control	5.2	Moderate
Alarm Advisor (Response to Plant Casualties)	5.1	Moderate
Pulverizer/Exhauster Replacement Part/Materials Advisor	5.1	Moderate
Sensor Management (Validation/Trending/Maintenance/Calibration)	5.1	Moderate
Turbine Steam Path Inspection	5.1	Moderate
Coal Preparation	4.9	Moderate
Consumables/Fuel Management Database/Advisor	4.9	Moderate
Predictive Maintenance - Life Extension	4.7	Moderate
Water Management	4.7	Moderate
Air Heater Operation Advisor	4.6	Moderate
Precipitator Energy Management and Optimization	4.6	Moderate
Systematic Trends	4.5	Lower
Wet/Dry FGD Design/Cost Estimating/Optimization	4.5	Lower
Management Decision Impact Evaluator	4.4	Lower
Optimization of Load Dispatching	4.4	Lower
Spare Parts Inventory and Ordering	4.4	Lower
Coal Gasification Process Selection	4.2	Lower
Turbine Performance Trend Analysis (Off-Line)	4.1	Lower
Expert Tuning System	4.0	Lower
Life Extension Advisor	3.9	Lower
Effects of Plant Ops on Boiler/Turbine Performance/Maintenance	3.8	Lower
Station Manager's Assistant	3.8	Lower
Fluidized Bed Boiler Selection and Optimization	3.7	Lower
Gasification Sulfur Removal/Recovery Selection and Optimization	3.7	Lower
Generalized Equipment and Diagnostics Shell	3.1	Lower

Note: These ratings are based on the following weightings -
Benefit/Cost - 40%
Need - 15%
Suitability - 15%
Availability - 15%
Development Time - 15%

- ability to encapsulate senior operators' expertise and serve as advisory systems,

- suitability of necessary hardware and software for the operating environment appropriate to the application, and

- ability to simulate actual plant operation.

Equipment Diagnostics. Prospective expert system applications for equipment diagnostics were identified according to the following considerations:

- availability of expertise to diagnose problems,

- appropriateness of diagnosis methodologies for translation to an expert system,

- significant number of potential users, and

- potential for incremental development approach, providing the opportunity for intermediate results.

Information Management. Prospective expert system applications for information management were identified according to the following considerations:

- requirements for integration of information from multiple sources (e.g., mainframe flat-files, station records, user input, drawings, manuals, etc.);

- ability to improve user access, productivity, and efficiency by reducing the time required to get results, reducing the required number of staff having programming skills, and making existing data available for analysis and decision-making through the development of an effective user interface that does not require specific knowledge of the databases involved; and

- complexity of the relationships between databases that require simultaneous evaluation.

Screening of Applications

All of the identified applications were subsequently screened by the Formal Working Group using separate task forces for each category to determine those applications having the greatest potential for expert systems development. Each of the identified prospective expert system were screened according to the following general criteria:

- assessment of applicability,

- impact upon plant and/or utility operations, and

- implementation costs.

The resulting lists of potential expert system applications (unranked) for plant operations, equipment diagnostics, and information management are presented in Tables 1, 2, and 3 (6). Some of the specific criteria used in the screening of identified applications to arrive at these lists include

- plant design and operating history
  - plant size and age
  - boiler, turbine, generator, and balance of plant design characteristics
  - existing plant operating conditions

- plant operating characteristics
  - plant operating mode (i.e., base, cycling, peaking, etc.)
  - fuel type
  - water/steam cycle configuration and physical characteristics
  - operating and control practices, including water chemistry

- availability of plant data
  - plant operating logs
  - historical plant performance data
  - generic industry data comparable to the available plant data

- impact upon plant operations
  - improved plant efficiency and availability through optimization of operating practices
  - improved operator understanding and training
  - improved understanding of trends in plant operating data
  - improved plant "memory" of operating problems

- impact upon equipment reliability (diagnostics)
  - improved plant availability through more thorough troubleshooting and corrective actions
  - reduction in equipment downtime through increased preventive maintenance
  - improved equipment maintenance through more thorough understanding of equipment operation and problem history
  - reductions in the risks of catastrophic equipment failure

- impact upon utility operations (information management)
  - more efficient operations and outage planning
  - improved plant budgeting
  - improved budget allocations based on priorities

- improved decision making
  - improved productivity by reducing the number of staff having programming skills required to access and integrate data.

Rating of Applications

A quantitative rating system was developed for ranking each of the forty-four potential expert system applications that were identified and survived the screening process. This system was used to evaluate the practicability and worth of each potential application by assessing ratings for the following five criteria: benefit/cost, need, suitability, availability, and development time. Each of the

Table 1

POTENTIAL EXPERT SYSTEM APPLICATIONS
FOR PLANT OPERATIONS

Air Heater Operation Advisor

Alarm Advisor (Response to Plant Casualties)

Boiler Feed Pump and Driver Condition Diagnostics

Boiler Thermal Performance

Coal Preparation

Combustion Process Performance Analysis

Cycle Chemistry Advisor

Demonstration of Live, Intelligent Data Base (Materials Experience)

Effects of Plant Ops on Boiler/Turbine Performance/Maintenance

Evaluating Fuel Quality Effects on Plant Performance, etc.

Expert Tuning System

Intelligent User Interface to Fuel Cost Model/Evaluator

Optimization of Load Dispatching

Precipitator Diagnostic Trending

Precipitator Energy Management and Optimization

Predictive Maintenance - On-Line Failure Prevention

Scrubber Operation and Control

Sensor Management (Validation/Trending/Maintenance/Calibration)

Station Manager's Assistant

Systematic Trends

Turbine Performance Trend Analysis (Off-Line)

Turbine-Generator Diagnostics (Balancing)

Unit Cycling Advisor

Wastewater Treatment Control

Wet FGD System Chemistry Control

Wet FGD System Water Balance

Table 2

POTENTIAL EXPERT SYSTEM APPLICATIONS
FOR EQUIPMENT DIAGNOSTICS

Boiler Tube Failure Inspection/Prediction

Coal Gasification Process Selection

Consumables/Fuel Management Data Base/Advisor

Demonstrate Delivery of Expert System with EPRI Diagnostic Manual

Equipment Overhaul Planning Data Base/Advisor

Fan Inspector

Fluidized Bed Boiler Selection and Optimization

Gasification Sulfur Removal/Recovery Selection and Optimization

Generalized Equipment and Diagnostics Shell

Life Extension Advisor

Management Decision Impact Evaluator

Painting and Coatings Advisor

Predictive Maintenance - Life Extension

Pulverizer/Exhauster Replacement Part/Materials Advisor

Turbine Steam Path Inspection

Unit Executive Advisor

Water Management

Wet/Dry FGD Design/Cost Estimating/Optimization

Table 3

POTENTIAL EXPERT SYSTEM APPLICATIONS
FOR INFORMATION MANAGEMENT

Interface/Format for Distributed Info Processing Between ESs

Repair Parts Management Data Base/Advisor

Spare Parts Inventory and Ordering

rating criteria, including a composite rating, was measured on a numerical scale ranging from 1 (low) to 10 (high).

The overall, composite rating for each application was obtained by calculating a weighted average of the ratings assessed for these five criteria, based upon the following formula:

Composite Rating =

0.4 * Benefit/Cost + 0.15 * (Need + Suitability + Availability + Development Time)

The sensitivity of this weighting upon the ultimate ranking of the potential expert system applications was also evaluated, as will be discussed later.

The ratings assessed for each of the potential expert system applications, listed in alphabetical order, are summarized in Table 4 (6).

The attributes and development of each of the rating criteria used in the ranking of the prospective expert system applications are summarized as follows.

Benefit/Cost. This rating criterion is a measure of the predicted payback of each potential expert system and is comprised of the ratio of the annual estimated return to the total estimated amortized cost per user. The total estimated cost is a function of the Development Cost, Users, Use, Hardware Cost, and Software Cost. Each of the cost ratings that were assigned to the potential expert system applications, along with the derived Benefit/Cost rating, is listed in Table 5. The bases for each of the factors involved in the derivation of the Benefit/Cost rating and included in Table 5 (6) are described as follows.

Development Cost per User. This rating factor was calculated based on the overall costs per user of a given application, which include development cost, potential use and number of potential users, normalized to a scale of one to ten. Each of these rating factors is defined as follows.

- Development Cost - This rating factor is a linear function of the estimated total development cost of a potential expert system application.
- Users - Individual expert systems, by virtue of their function, will have different bases of potential users. Some applications will typically be used at the corporate level, while others will be used

Table 4

PRIORITIZATION RATING MATRIX

Application	Benefit/Cost	Need	Suitability	Availability	Dev. Time	Rating
Air Heater Operation Advisor	2	4	4	9	8	4.6
Alarm Advisor (Response to Plant Casualties)	1	9	9	7	6	5.1
Boiler Feed Pump and Driver Condition Diagnostics	4	4	8	7	6	5.4
Boiler Thermal Performance	3	6	7	9	7	5.6
Boiler Tube Failure Inspection/Prediction	3	4	8	10	9	5.9
Coal Gasification Process Selection	1	7	5	5	8	4.2
Coal Preparation	5	3	4	5	7	4.9
Combustion Process Performance Analysis	3	5	8	9	7	5.6
Condenser/Feedwater Heater Diagnostics	3	5	5	10	9	5.6
Consumables/Fuel Management Database/Advisor	2	7	5	7	8	4.9
Cycle Chemistry Advisor	2	8	10	9	7	5.9
Demonstrate Delivery of Expert System with EPRI Diagnostic Manual	2	10	10	10	9	6.7
Demonstration of Live, Intelligent Database (Materials Experience)	2	10	7	7	7	5.5
Effects of Plant Ops on Boiler/Turbine Performance/Maintenance	2	5	5	5	5	3.8
Equipment Overhaul Planning Database/Advisor	2	9	10	9	7	6.1
Expert Tuning System	2	4	6	5	6	4.0
Fan Inspector	2	6	9	8	7	5.3
Fluidized Bed Boiler Selection and Optimization	1	4	5	6	7	3.7
Gasification Sulfur Removal/Recovery Selection and Optimization	1	5	4	6	7	3.7
Generalized Equipment and Diagnostics Shell	4	2	5	1	2	3.1
Intelligent User Interface to Fuel Cost Model/Evaluator	4	10	9	10	9	7.3
Interface/Format for Distributed Info Processing Between ESs	4	8	10	8	6	6.4
Life Extension Advisor	3	4	4	5	5	3.9
Management Decision Impact Evaluator	3	5	6	5	5	4.4
Optimization of Load Dispatching	2	5	6	7	6	4.4
Painting and Coatings Advisor	2	8	5	8	9	5.3
Precipitator Diagnostic Trending	3	6	8	8	7	5.6
Precipitator Energy Management and Optimization	2	5	4	8	8	4.6
Predictive Maintenance - Life Extension	2	4	7	8	7	4.7
Predictive Maintenance - On-Line Failure Prevention	4	5	7	7	6	5.4
Pulverizer/Exhauster Replacement Part/Materials Advisor	4	5	5	6	7	5.1
Repair Parts Management Database/Advisor	2	8	8	9	7	5.6
Sensor Management (Validation/Trending/Maintenance/Calibration)	3	8	5	6	7	5.1
Spare Parts Inventory and Ordering	2	6	3	8	7	4.4
Station Manager's Assistant	3	3	4	4	6	3.8
Systematic Trends	3	3	3	8	8	4.5
Turbine Performance Trend Analysis (Off-Line)	3	2	6	5	6	4.1
Turbine Steam Path Inspection	3	5	5	8	8	5.1
Turbine-Generator Diagnostics (Balancing)	4	7	8	7	6	5.8
Unit Cycling Advisor	2	8	9	8	7	5.6
Unit Executive Advisor	3	6	9	10	7	6.0
Water Management	3	5	4	8	6	4.7
Wet FGD System Chemistry Control	1	10	8	7	7	5.2
Wet/Dry FGD Design/Cost Estimating/Optimization	1	3	5	10	9	4.5

Note: These ratings are based on the following weightings -

Benefit/Cost - 40%
Need - 15%
Suitability - 15%
Availability - 15%
Development Time - 15%

Table 5

COST/BENEFIT EVALUATION RATING MATRIX

Applications	Development Cost	Users	Use	Costs per User					Return	Benefit/Cost
				Development	Hardware	Software	Training	Total		
Air Heater Operation Advisor	3	9	2	1	2	1	1	2	2	2
Alarm Advisor (Response to Plant Casualties)	2	9	1	1	3	1	4	3	1	1
Boiler Feed Pump and Driver Condition Diagnostics	2	9	2	1	1	1	1	1	2	4
Boiler Thermal Performance	3	9	1	2	3	1	6	4	6	3
Boiler Tube Failure Inspection/Prediction	1	9	1	1	1	1	4	3	4	3
Coal Gasification Process Selection	3	2	3	3	1	1	1	2	1	1
Coal Preparation	4	2	3	3	2	1	1	2	5	5
Combustion Process Performance Analysis	6	9	1	3	2	2	2	3	4	3
Condenser/Feedwater Heater Diagnostics	3	9	1	2	1	1	1	2	2	2
Consumables/Fuel Management Database/Advisor	5	4	2	3	1	1	2	2	2	2
Cycle Chemistry Advisor	3	9	2	1	1	1	3	2	2	2
Demonstrate Delivery of Expert System w/ Diagnostic Manual	1	5	2	1	1	1	1	1	1	2
Demonstration of Live, Intelligent Database (Materials Experience)	2	4	2	1	2	1	1	2	2	2
Effects of Plant Ops on Boiler/Turbine Performance/Maintenance	7	9	1	4	5	5	2	6	6	2
Equipment Overhaul Planning Database/Advisor	6	9	2	2	8	5	4	7	6	2
Expert Tuning System	3	4	2	2	4	1	1	3	3	2
Fan Inspector	1	9	1	1	1	1	1	1	1	2
Fluidized Bed Boiler Selection and Optimization	3	2	1	8	1	1	1	3	1	1
Gasification Sulfur Removal/Recovery Selection and Optimization	2	2	1	5	1	1	1	2	1	1
Generalized Equipment and Diagnostics Shell	6	4	2	4	1	0	1	2	4	4
Intelligent User Interface to Fuel Cost Model/Evaluator	1	2	4	1	1	1	1	1	2	4
Interface/Format for Distributed Info Processing Between ESs	2	2	4	1	1	1	1	1	2	4
Life Extension Advisor	4	4	2	3	1	1	1	2	3	3
Management Decision Impact Evaluator	6	2	3	5	1	1	1	2	3	3
Optimization of Load Dispatching	8	9	1	4	5	3	2	5	5	2
Painting and Coatings Advisor	1	6	2	0	1	1	1	1	1	2
Precipitator Diagnostic Trending	2	7	2	1	3	1	1	2	3	3
Precipitator Energy Management and Optimization	4	7	2	1	2	1	1	2	2	2
Predictive Maintenance - Life Extension	3	9	1	2	1	1	1	2	2	2
Predictive Maintenance - On-Line Failure Prevention	1	9	1	1	1	1	1	2	4	4
Pulverizer/Exhauster Replacement Part/Materials Advisor	2	7	3	0	1	1	1	1	2	4
Repair Parts Management Database/Advisor	4	9	4	1	6	5	3	6	5	2
Sensor Management (Validation/Trending/ Maintenance/Calibration)	6	4	2	4	4	1	1	3	4	3
Spare Parts Inventory and Ordering	4	4	2	3	5	2	1	4	3	2

Table 5, Cont.

Applications	Development Cost	Users	Use	Costs per User: Development	Hardware	Software	Training	Total	Return	Benefit/Cost
Station Manager's Assistant	4	4	3	2	1	1	1	2	3	3
Systematic Trends	4	9	1	2	3	1	1	2	3	3
Turbine Performance Trend Analysis (Off-Line)	6	9	1	3	3	1	1	3	5	3
Turbine Steam Path Inspection	3	6	1	3	1	1	2	2	3	3
Turbine-Generator Diagnostics (Balancing)	2	9	2	1	1	1	1	1	2	4
Unit Cycling Advisor	3	5	2	2	10	1	3	6	7	2
Unit Executive Advisor	2	4	1	3	1	1	4	3	5	3
Water Management	3	4	3	1	2	1	1	2	3	3
Wet FGD System Chemistry Control	6	2	4	4	7	9	8	10	5	1
Wet/Dry FGD Design/Cost Estimating/Optimization	3	1	2	8	1	1	1	3	1	1

at the station or unit levels. The level of use will be the major factor in the ultimate probable number of users for a given expert system application. Furthermore, some applications, although developed for fossil-fired plants, may also be applicable to nuclear plants, thereby increasing the base of potential users. Those expert system applications which have the largest number of potential users will have increased benefits and possibly greater payback to the investment in development.

This rating factor is proportionate to the total number of possible users for a given expert system application.

- Use - This rating factor reflects an assessment of the share of the potential market that will be captured by a given expert system application. This rating factor is proportionate to the percentage of the total number of possible users (designated by the rating factor, Users), that are expected to make productive use of a given expert system application.

Hardware Costs per User. Even when all costs are funded by EPRI for the complete development of a given expert system application, there are tangible costs associated with implementing the completed expert system at individual utilities, stations, and/or units that must be borne by the users. These implementation costs partially offset any gross returns in performance and/or productivity that may be realized through the use of the expert system. One of these implementation costs is any hardware (microcomputer, minicomputer, or LISP workstation; data acquisition systems; specialized I/O interfaces; etc.) that will be required to use a specific expert system.

The rating factor for hardware costs is a linear function of the estimated total installed cost of any dedicated hardware that is expected to be required for implementation of a given potential expert system application by a single user.

Software Costs per User. Another of the implementation costs that may be associated with a given expert system application is that for any software required to run and/or modify the expert system. The rating factor for software costs is a linear function of the estimated total purchase cost of any dedicated software that is expected to be required for implementation of a given potential expert system application by a single user. The scale for this rating factor is identical to that for Hardware Costs per User, with the exception that a rating of zero being used to denote the expectation that no software would be required to be purchased by the user.

Training Costs per User. This factor rates the training cost of an application on a per user basis. If the cost of the training was at or below 200 man-hours, a rating of 1 was given. If the training required over 1800 man-hours, a rating of 10 was used.

Total Cost per User. This parameter was calculated based upon a weighted average of the development, hardware, software, and training costs per user.

Return. The return on an application is the anticipated cost savings that an application can produce. This includes factors such as improved availability, reduced operating and maintenance labor, and fuel costs. This factor was calculated on a per user basis. An application that saved a user less than $50,000 dollars per year was given a ranking of 1. An application that saved more than $500,000 per year was given a rating of 10.

Benefit/Cost. This parameter is the normalized ratio of the application's return to the total cost.

Need. This factor evaluates the relative urgency of developing an expert system. If it is a system whose immediate implementation will result in large cost savings, the need is considered high. Likewise an expert system whose development is attractive but whose implementation produces no immediate benefit has little need.

Suitability. Prior to proposing an expert system application for serious consideration and evaluation, several simple questions can be asked of the proposed application to determine whether or not this application is worth pursuing. If one or more of these questions cannot be answered, then it is likely that it will be either difficult to develop the expert system or difficult to justify its use when completed. The numerical value assigned to this was based on the number of questions that were answered correctly. An application that meets all of this criteria was given a rating of 10. The screening questions used to assess the suitability of each potential application for development as an expert system include

- Are there recognized experts in the field or on this topic? The lack of experts means either there is no need to develop the application (i.e., everyone is an expert) or the subject matter is not understood well enough for someone to become an expert.

- Are the experts better than the amateurs? In many areas, this rule is not often the case; however, in many different areas of utility applications, experts do have their place and they are frequently used by utilities for assistance.

- Does the task take less than a day for an expert to do (less than two hours is preferred)? A complicated task that will take an expert several days to analyze will be quite difficult to develop on a computer.

- Is the task primarily cognitive? If the task doesn't require factual knowledge, it will be difficult to model an expert system around it.

- Does the task require use of "deep knowledge"? This may also be hard to express in an expert system since it is reasoning that is hard to explain in simplistic terms.

- Is this skill routinely taught to neophytes? If it is, there will be many experts in a short amount of time and consequently no need for an expert system.

- Is there a high payoff? An application that can produce a high payoff is obviously more worthwhile.

Availability of Expertise. This factor weighs the availability of experts to assist in developing the experts. If there is a lack of recognized experts in the field, an expert system developed for this type of application would lack some credibility. If the subject matter is vague or relatively new, the factor was given a lower value. For example, equipment that is used widespread throughout the industry such as turbines and boilers have plenty of experts available and consequently can have this expertise more readily converted into an expert system.

Development Time. This factor evaluates development time for an expert system and was defined to be inversely proportional to the estimated development time for the expert system. The rankings are based on the fact that if it takes a long time to develop an application, the program may become obsolete by the time it is completed.

Sensitivity Analysis

The composite ratings of each application are included in Table 4 (6). These ratings are based upon the preceding equation, which assigns the greatest weight to the benefit/cost ratio of the applications. The benefit/cost should logically be the primary driving force towards implementation; therefore, the rankings obtained by this weighting should receive the most serious consideration in the selection of potential applications for development. The relative impacts of the other rating criteria upon the ranking of the applications were assessed by performing a sensitivity analysis on the weightings assigned to the individual rating criteria. Since the values and weightings assessed to the rating criteria for each of the potential expert system applications are primarily subjective, this sensitivity analysis serves to identify inconsistent ratings as well as applications that might warrant

higher ranking based upon other considerations. The sensitivities of assigning the greatest weight to each of the rating criteria, as well as that of weighting each of the criteria equally, were evaluated. The results of this analysis are presented in Table 6 (6).

Prioritization

The sensitivity analysis summarized in Table 6 supported breaking the potential expert system applications into groups of relative priorities. Three priority groups were identified: higher, intermediate, and lower. Although the absolute ranking of potential applications would differ depending upon the rating criteria weighting used, the grouping would be relatively unaffected. The final ranking of the applications is summarized in Table 7 (6). Within each of the three priority groupings, the priority of individual applications was considered to be essentially equal.

Expert Systems Projects

Based on the application rating and prioritization, nineteen expert system applications distributed among three categories (plant operations, equipment diagnostics, and information management) were identified for near-term development and implementation. These projects, plus one study that applies to all expert system development efforts, were selected based on an evaluation of 50 utility-suggested applications that addressed criteria including need, suitability, and cost-benefit ratio. The payback period of these applications ranges from 1 to 4 years. The estimated total funding level for these projects is $7,000,000 over the next 5 to 7 years. These projects, which were evaluated as having the highest priorities of the suggested applications, are listed in Figure 5 (6), which also presents the anticipated time line for each project. Descriptions of each of the projects are contained in the R&D Plan report.

Figure 5 illustrates the planned phasing for the initiation and completion of each expert system development project. The phasing of these projects is consistent with the priority groupings. This phasing also takes into account the need to maintain a broad range of applications in development, and therefore does not follow precisely the numerical rankings. Within the margin of error inherent to the unavoidable subjectivity of some of the rating criteria, the project phasing is also considered to be consistent with these rankings. This plan calls for the initiation of six expert system projects in the first year followed by the initiation of an average of three new expert system applications per year over the next four years. This program will result in a steady growth in the number of expert system applications

Table 6

SENSITIVITY ANALYSIS

Application	Ratings							
	Base	A	B	C	D	E	F	G
Intelligent User Interface to Fuel Cost Model/Evaluator	7.3	8.8	8.6	8.8	8.6	8.4	7.4	7.8
Demonstrate Delivery of Expert System with EPRI Diagnostic Manual	6.7	8.7	8.7	8.7	8.4	8.2	6.8	7.5
Interface/Format for Distributed Info Processing Between ES's	6.4	7.4	7.9	7.4	6.9	7.2	6.6	7.2
Equipment Overhaul Planning Database/Advisor	6.1	7.8	8.1	7.8	7.3	7.4	6.3	7.0
Unit Executive Advisor	6.0	6.8	7.5	7.8	7.0	7.0	6.1	6.7
Boiler Tube Failure Inspection/Prediction	5.9	6.1	7.1	7.6	7.4	6.8	5.7	6.3
Cycle Chemistry Advisor	5.9	7.4	7.9	7.7	7.2	7.2	6.1	6.9
Turbine-Generator Diagnostics (Balancing)	5.8	6.6	6.8	6.6	6.3	6.4	5.9	6.3
Boiler Thermal Performance	5.6	6.3	6.6	7.1	6.6	6.4	5.6	6.0
Combustion Process Performance Analysis	5.6	6.1	6.8	7.1	6.6	6.4	5.6	6.1
Condenser/Feedwater Heater Diagnostics	5.6	6.1	6.1	7.3	7.1	6.4	5.4	5.6
Precipitator Diagnostic Trending	5.6	6.3	6.8	6.8	6.6	6.4	5.6	6.1
Repair Parts Management Database/Advisor	5.6	7.1	7.1	7.4	6.9	6.8	5.8	6.3
Unit Cycling Advisor	5.6	7.1	7.4	7.1	6.9	6.8	5.7	6.4
Demonstration of Live, Intelligent Database (Materials Experience)	5.5	7.5	6.7	6.7	6.7	6.6	5.6	6.0
Boiler Feed Pump and Driver Condition Diagnostics	5.4	5.4	6.4	6.1	5.9	5.8	5.3	5.9
Predictive Maintenance - On-Line Failure Prevention	5.4	5.6	6.1	6.1	5.9	5.8	5.4	5.7
Fan Inspector	5.3	6.3	7.1	6.8	6.6	6.4	5.3	6.1
Painting and Coatings Advisor	5.3	6.8	6.1	6.8	7.1	6.4	5.2	5.4
Wet FGD System Chemistry Control	5.2	7.5	7.0	6.7	6.7	6.6	5.4	6.0
Alarm Advisor (Response to Plant Casualties)	5.1	7.1	7.1	6.6	6.3	6.4	5.3	6.0
Pulverizer/Exhauster Replacement Part/Materials Advisor	5.1	5.3	5.3	5.6	5.8	5.4	4.9	5.1
Sensor Management (Validation/ Trending/Maintenance/Calibration)	5.1	6.4	5.6	5.9	6.1	5.8	5.1	5.2
Turbine Steam Path Inspection	5.1	5.6	5.6	6.4	6.4	5.8	5.0	5.2
Coal Preparation	4.9	4.4	4.6	4.9	5.4	4.8	4.6	4.6
Consumables/Fuel Management Database/Advisor	4.9	6.1	5.6	6.1	6.4	5.8	4.8	5.0

Table 6, Cont.

Application	Ratings							
	Base	A	B	C	D	E	F	G
Predictive Maintenance - Life Extension	4.7	5.2	6.0	6.2	6.0	5.6	4.6	5.2
Water Management	4.7	5.2	4.9	5.9	5.4	5.2	4.7	4.7
Air Heater Operation Advisor	4.6	5.1	5.1	6.3	6.1	5.4	4.4	4.6
Precipitator Energy Management and Optimization	4.6	5.3	5.1	6.1	6.1	5.4	4.4	4.6
Systematic Trends	4.5	4.5	4.5	5.8	5.8	5.0	4.3	4.3
Wet/Dry FGD Design/Cost Estimating/Optimization	4.5	5.0	5.5	6.7	6.5	5.6	4.2	4.7
Management Decision Impact Evaluator	4.4	4.9	5.1	4.9	4.9	4.8	4.4	4.7
Optimization of Load Dispatching	4.4	5.2	5.4	5.7	5.4	5.2	4.4	4.8
Spare Parts Inventory and Ordering	4.4	5.4	4.7	5.9	5.7	5.2	4.4	4.3
Coal Gasification Process Selection	4.2	5.7	5.2	5.2	5.9	5.2	4.0	4.4
Turbine Performance Trend Analysis (Off-Line)	4.1	3.8	4.8	4.6	4.8	4.4	3.8	4.4
Expert Tuning System	4.0	4.5	5.0	4.7	5.0	4.6	3.8	4.4
Life Extension Advisor	3.9	4.2	4.2	4.4	4.4	4.2	3.9	4.0
Effects of Plant Ops on Boiler/ Turbine Performance/Maintenance	3.8	4.6	4.6	4.6	4.6	4.4	3.8	4.1
Station Manager's Assistant	3.8	3.8	4.0	4.0	4.5	4.0	3.5	3.8
Fluidized Bed Boiler Selection and Optimization	3.7	4.5	4.7	5.0	5.2	4.6	3.5	4.0
Gasification Sulfur Removal/ Recovery Selection and Optimization	3.7	4.7	4.5	5.0	5.2	4.6	3.6	3.9
Generalized Equipment and Diagnostics Shell	3.1	2.6	3.4	2.4	2.6	2.8	3.1	3.4

Note: These ratings are based on the following weightings -

	Base	A	B	C	D	E	F	G
Benefit/Cost -	40%	15%	15%	15%	15%	20%	40%	30%
Need -	15%	40%	15%	15%	15%	20%	20%	15%
Suitability -	15%	15%	40%	15%	15%	20%	15%	30%
Availability -	15%	15%	15%	40%	15%	20%	20%	15%
Development Time -	15%	15%	15%	15%	40%	20%	5%	10%

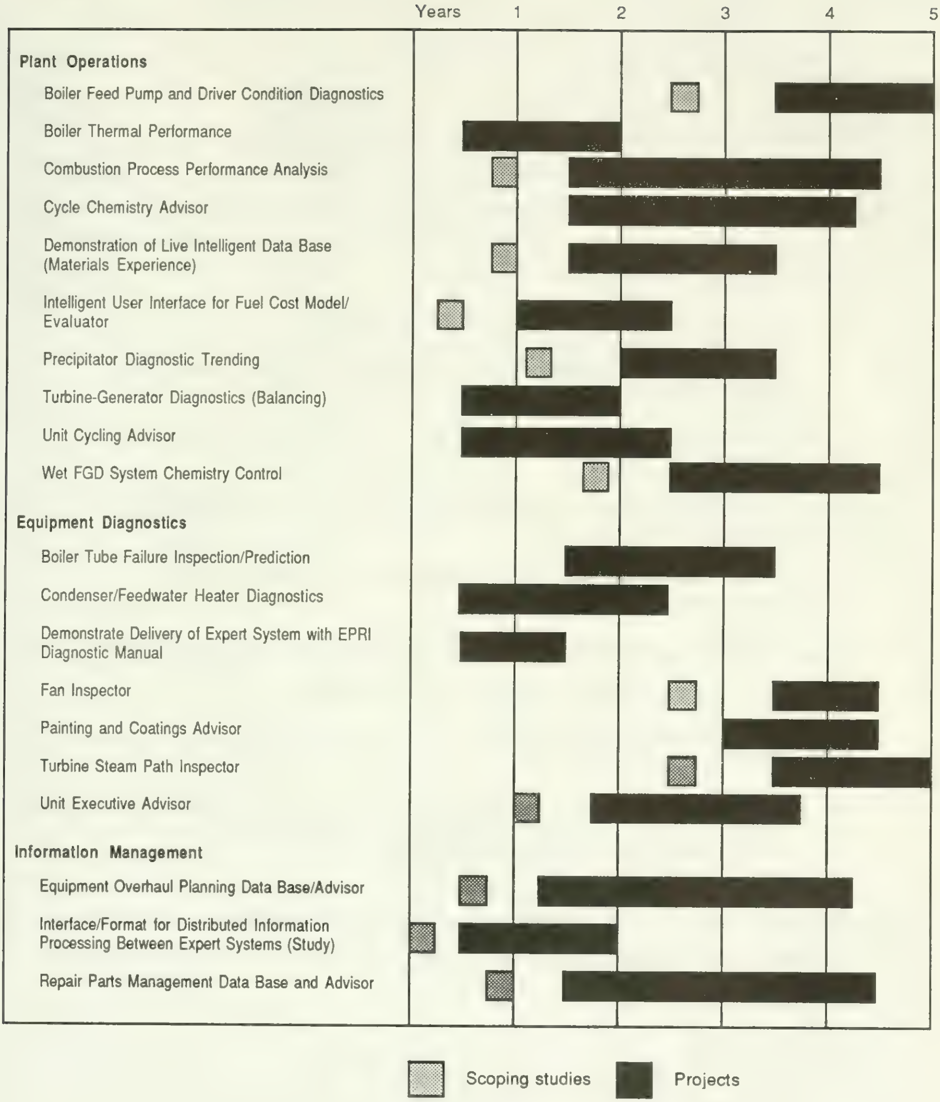


Figure 5—Expert Systems Time Line

in use by the utility industry. Once expert systems technology is better understood and a development framework is established, each EPRI R&D program for fossil fuel power plants may supplement other projects with additional expert systems as an alternate means of technology transfer.

Expert Systems Development Approach

The R&D plan report specifies a standardized approach for all EPRI expert systems development projects for fossil fuel power plants to ensure a similar "look and feel" and consistency of approach while maximizing productivity. This approach includes the following key features:

- utility participation in the development process;
- incorporation of a standardized interface/format for information transfer between expert systems;
- use of established knowledge bases from existing EPRI R&D work and human expertise;
- use of commercially available, PC-based expert systems shells for application development;
- expert system-specific quality assurance and documentation requirements;
- application design for use by most utilities without modification;
- adequate provisions for application maintenance;
- documentation and technology transfer requirements to assure adequate user training; and
- submittal of commercialization plans that provide mechanisms to ensure long-term application support with proposals for application development by prospective contractors.

The R&D plan report also includes detailed discussions on the selection of software, project planning, and project management for expert systems development.

Prospective Expert Systems Network

Implementation of this approach will result in the development of a series of stand-alone expert systems, each having the built-in capability to link to one another through a computer network. Installation of most of the expert systems to be developed in a plant will create a comprehensive, integrated expert systems network that will assist operators in optimizing most critical aspects of plant operation. Figure 6 (4) illustrates this network, along with the direction(s) of information flow from one expert systems application to the next. Information flow in the

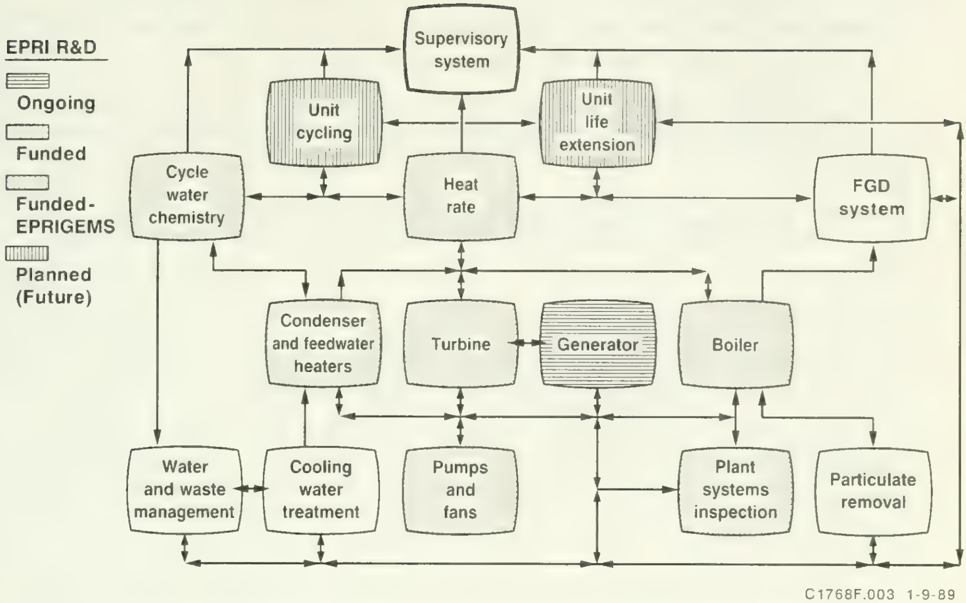


Figure 6—Prospective Expert Systems Network for Fossil Fueled Power Plants

network will be hierarchical, with results from equipment-oriented expert systems funneling up to process-oriented expert systems, which in turn provide information on overall plant status to a single supervisory system (7). Thus the expert system applications to be developed will simultaneously

- provide focused applications suitable for stand-alone implementation at plants having a specialized need or pursuing incremental implementation of expert systems technology, and
- provide an integrated delivery vehicle so that plants implementing the full range of expert systems technology will not burden operators with multiple sources of redundant information.

CONCLUSIONS

Implementation of expert systems technology in the utility industry offers the potential for improvements in plant reliability, availability, and heat rate; equipment life; and personnel productivity across a variety of applications. These potential benefits coincide with current industry priorities, causing the implementation of expert system technology in the utility industry to take on a sense of urgency.

The implementation of any new technology must be accomplished with a certain degree of caution. Premature application of even the best of new technologies often results in unanticipated problems that detract from the industry perception of the potential benefits that may be realized. Conversely, protracted delays in the implementation of a new technology breed a perception that the technology is not ready for industry implementation. The best approach, therefore, is to implement the technology in selected, low-risk, but important, applications that may be developed and proven in a relatively short time frame. These "quick successes" serve to demonstrate the technology and stimulate interest in the development of more advanced, higher risk applications. Also, to fully realize the benefits of the technology, education on the part of the utilities is needed. These smaller applications will contribute towards industry education in expert systems technology.

A variety of expert system development tools, or software shells, have recently become available, including many that have been specifically designed for use on personal computers. The advent of these tools has placed the development of straightforward expert systems of limited scope within the capabilities of most utilities. Indeed, several utilities have already undertaken the development of some limited expert system applications for the primary purpose of demonstrating the technology. These efforts are commendable and should be encouraged to continue. These demonstrations may be successful in paving the way for gradual implementation

of expert system technology within the individual utilities that are involved, but will not cause the majority of utilities to implement expert system technology in a reasonable time frame, however. Furthermore, those prospective expert system applications having the highest potential benefit generally also have development costs that are too high to be borne by any one utility. Consequently, the full benefits of expert system technology are unlikely to be realized if left to the efforts of individual utilities.

Equipment manufacturers, consultants, and A/Es have also realized the potential benefits of expert system technology. Several expert system applications have been developed to the commercialization stage and many more are in various stages of development. Although there appears to be a fair amount of activity on the surface, in reality most of these companies are proceeding at a relatively minimal level of effort due to the high cost of developing expert system applications and the uncertain market for the licensing or purchase of this software and related services by utilities. Realization of the aforementioned "quick successes" by a variety of selected expert system applications in the utility industry will create a market demand for expert system applications that will stimulate accelerated development of worthwhile applications by equipment manufacturers, consultants, and A/Es.

Development of a productive expert system application requires some specialized expertise in the field of artificial intelligence (AI) as well as expertise in the chosen field of application. The availability of personal computer-based expert system shells and the inherent attractiveness of expert system technology create an environment conducive to the development of many small-scale expert systems by a variety of people having varying amounts of expertise. Given the relative newness of AI as a recognized field of study, most people experimenting with expert system applications are not likely to have in-depth knowledge of this technology. Therefore, small-scale development efforts are not likely to take full advantage of available AI technology. For this reason, there is a danger inherent to undirected industry implementation of expert system technology in that some of these first applications may not be successful, thereby retarding the momentum for implementation of the technology in general.

In time, the level of AI expertise required to develop a successful, "intelligent" expert system will diminish, primarily as a result of the development of smarter, more powerful software shells and the faster hardware necessary to make these more sophisticated shells practicable for implementation. These advances in technology will place the development of productive, but still smaller-scale, expert system

applications within the capabilities and resources of most utilities. In the interim, there is a need for EPRI to take the lead in the implementation of expert system technology in the utility industry. This will be accomplished by way of a structured, integrated program that will produce several "quick successes" across a variety of fields of application, followed by an organized program of accelerated development of more comprehensive applications. This program will have the following parallel objectives:

- clarify general concepts and demonstrate the benefits of expert systems technology to the utility industry using small and easily understandable applications,
- build the necessary tools to enable member utilities to realize the benefits of expert systems technology,
- develop basic application-specific expert system platforms that may be easily customized to create plant-specific expert systems for any given plant,
- develop high-benefit applications that are unlikely to be developed by individual utilities or companies because of high development costs,
- develop the necessary guidelines and interfaces for expert systems development that will enable utilities to mix and match various EPRI-, utility-, and commercially-developed expert systems on available utility computer facilities,
- create technology transfer vehicles, including tools, that will facilitate expeditious, widespread usage of expert systems technology, including the independent, unassisted development of successful, small-scale expert system applications by utilities, and
- develop generic guidelines to the industry that will provide direction and consistency to industry-wide expert system implementation efforts.

Expert systems having the greatest potential benefits for fossil fuel power plants have been identified, screened, rated, ranked, and prioritized. The selected applications constitute a long range program for EPRI-assisted implementation of expert systems technology.

EPRI is beginning to implement this R&D plan. Eight expert systems for fossil fuel power plants are presently under development by EPRI (8). Working with technical experts in the utility industry, these systems are being developed and tested in an off-line mode; later, several of these systems will be installed on-line in power plant control rooms, where they will undergo further verification and validation. These eight projects include

- Boiler Tube Failure Diagnosis System,
- Electrical Generator Monitoring System,
- Turbine Condition Monitoring System,
- Heat Rate Degradation Advisor,
- Condenser and Feedwater Heater Advisors,
- Cycle Water Chemistry and Demineralizer Operation Advisor,
- Cycling Advisor, and
- Plant Systems Inspection Advisor.

Development of several of the other expert systems in the R&D plan is expected to begin in the near future.

REFERENCES

1. J. Martin. "Modeling Technology: Trends for the Early 1990's." PC Week. Ziff-Davis Publishing Company.

2. National Electric Reliability Council.

3. The Adequacy of U.S. Electricity Supply Through the Year 2000. AER Company. May 1987.

4. Executive Summary: R&D Plans for Advanced Computer and Control Technologies. EPRI RP2819-5. Electric Power Research Institute. Palo Alto, California. November 1988.

5. J. Martin. Modeling Technology: Trends for the Late 1990's." PC Week. Ziff-Davis Publishing Company., October 24, 1988.

6. Expert Systems Technology Assessment and R&D Plan for Fossil Plant Applications. EPRI RP2819-5. Electric Power Research Institute. Palo Alto, California. April 1989.

7. Sopocy, D. M., Glazer, A. R., Rice, J. K., and Jonas, O., "Expert Systems to Improve Plant Operability." Proceedings of the American Power Conference, Volume 50. Illinois Institute of Technology. Chicago, Illinois. April 1988.

8. Valverde A., Jr., L. J.; Gehl, S. M., Armor, A. F.; Scheibel, J.; and Divakaruni, S. M.; "Fossil Power Plant Applications of Expert Systems: An EPRI Perspective." Presented at the 51st American Power Conference, Chicago, Illinois. April 24 - 26, 1989.

An Expert System-based, On-Line Rotor Crack Monitor for Utility Steam Turbines

JOHN R. SCHEIBEL
Electric Power Research Institute
3412 Hillview Avenue
Palo Alto, California 94304, USA

IMDAD IMAM
Mechanical Systems Dynamics Project

THOMAS G. EBBEN
GE Corporate Research and Development
Schenectady, New York, USA

RICK BLOMGREN
Florida Power and Light
Fort Lauderdale, Florida, USA

ABSTRACT

A steam turbine vibration monitoring system is described that uses a rule-based expert system for data review and fault diagnosis. Steady-state, coast-down, and steam temperature transient vibration signature techniques used by the monitor to detect transverse rotor cracks are summarized. A histogram technique for enhancing the initial appearance of a shallow crack 2/rev response is presented. The use of an expert system to fully automate diagnosis of turbine faults is discussed. Rotor crack and misalignment diagnostic rules are outlined.

INTRODUCTION

The EPRI turbomachinery diagnostic monitoring program has primarily focused on the development of computer-based vibration spectral data collection and analysis systems as a replacement for the simple alarm function of overall vibration supervisory instrumentation. These advanced monitoring systems, capable of a range of data processing techniques and displays, require expert review to diagnose the fault type and severity. In a typical utility plant setting, such a specialized machinery analyst is not routinely available. Only infrequently when a problem exists is an expert evaluation of the vibration

Vibration signatures can be ambiguous and equipment dependent, and therefore insufficient to define a specific fault. Figures 1 and 2 indicate similar vibration responses that are associated with a variety of faults (*2*). For example, a 1/rev vibration may be due to a change in unbalance force, system stiffness, rotating speed, or system damping. A 2/rev vibration may be produced by a change in rotor and/or bearing stiffness, or by misalignment of the rotor at the bearings. To mistake high vibration from a rotor crack for unbalance or misalignment could be a costly error.

While vibration signature data are analyzed using various signal processing techniques to help discriminate between fault types, other types of data may also be required. For instance, rotor position, bearing temperature, or performance data may contain clues necessary to narrow the number of probable faults. An expert system provides a framework for a diagnostic process that evaluates the evidence from a range of sensors.

KNOWLEDGE BASE RELIABILITY AND GROWTH

A properly developed and maintained expert system should provide the user with a consistent level of technical review. Additional rules derived from technical and experiential resources, including a variety of experts, are readily incorporated into the software. As new rules are

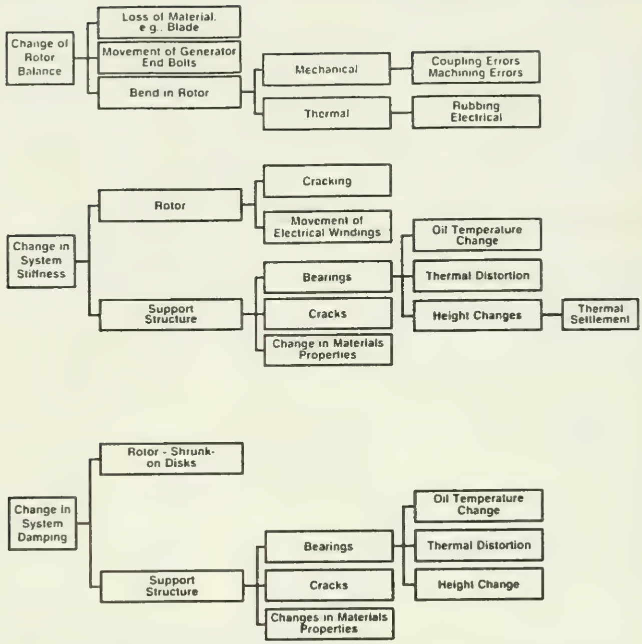


Figure 1—Possible causes of 1/rev vibration in rotating machinery.

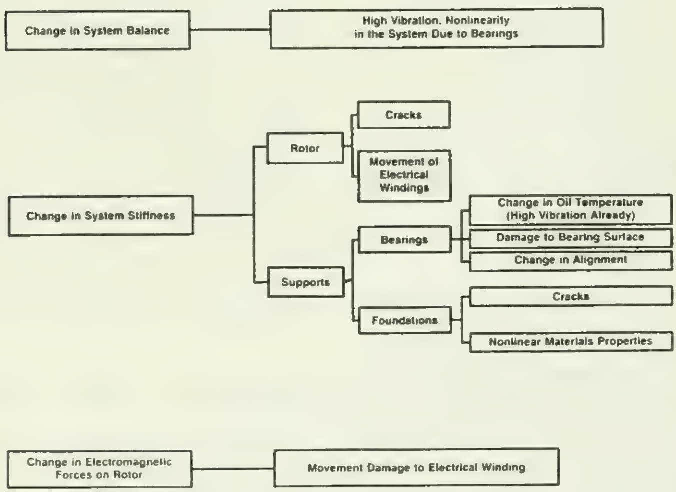


Figure 2—Possible causes of 2/rev vibration in rotating machinery.

added or obsolete rules discarded, the expert system automatically adjusts the affected logic trees. This minimizes programming costs for expanding or customizing the knowledge base. Of course, any expert system is ultimately bounded by the depth of knowledge and experience available.

DIAGNOSTIC PROCESS EFFECTIVENESS

An expert system is most suitable for a complex diagnostic process considering many fault types with multiple symptoms. Efficient expert system logic asks only relevant questions and requires that a fact be given only once. Once a diagnosis is completed, an operator need not be familiar with software details to review the reasoning process immediately. Recently several notable expert systems for rotating machinery diagnostics have been developed (*3,4,5*). Most of these require that an operator supply simple answers to information requests. This can be tedious, especially if data collection and analysis is involved. In order to eliminate this operator interface, numeric sensor data acquired by a monitoring system must be converted to symbolic statements required by the expert system.

STEAM TURBINE CONDITION MONITOR

The expert system described here is housed in a separate computer and acquires on-line turbine generator condition data directly from a microprocessor-based vibration signature analysis monitor. Vibration, temperature, shaft position, and phase angle are monitored during steady-state and coast-down operation. Vibration data are presented in forms familiar to analysts, including trend, waterfall, and Bode plots. An HP computer performs the data collection, processing, and numeric analyses, while a Compaq 286 performs the symbolic, expert system diagnosis. Figure 3 indicates the data link between the two computers. This system was installed at Florida Power and Light utility in 1986 and has been operational since then. The field testing results from this system are described later in the paper.

ROTOR CRACK SIGNATURES

The monitor uses several signature analysis approaches to indicate the initial propagation of a transverse rotor crack. This failure, oriented circumferentially and radially, is the most common turbine generator rotor crack type. Crack vibration signatures can be misdiagnosed as an unbalanced or misaligned rotor. As a result, cracks may grow to dangerous proportions,

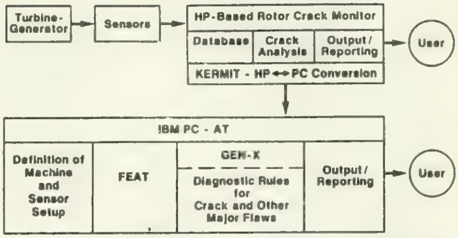


Figure 3—Turbine monitoring system uses two coupled computers to perform numeric monitoring and symbolic diagnostic functions.

threatening catastrophic equipment failure. In the past, utilities have used the coast-down and temperature-transient approaches to detect cracking. To further enhance crack signals produced by rotor stiffness asymmetry and suppress background vibration noise, a histogram technique was developed.

COAST-DOWN ANALYSIS

The coast-down analysis, successfully demonstrated by the Central Electricity Generating Board in England, detects changes in the absolute vibration spectrum as the machine decelerates through critical rotor speeds (*6*). Transient data are examined for these telltale crack signatures (*7,8,9*):

- 2/rev at one-half critical speed
- 3/rev at one-third critical speed
- Double critical 1/rev signature
- Reduction of critical speed
- Persistence of resonance at critical speeds

This approach has several drawbacks: it has limited sensitivity, only detects cracks with depths exceeding 25% of the rotor diameter, and disrupts electricity production while the machine is run down. Coast-down data shown in Figure 4 from one utility cracked rotor incident indicate the 2/rev and 3/rev responses (*11*).

TEMPERATURE-TRANSIENT ANALYSIS

U. S. utilities have used the temperature-transient method since 1974. A rapid steam temperature change induces thermal stresses that open cracks, causing asymmetrical rotor flexibility. The initial increase in rotor 2/rev vibration slowly reduces as the rotor temperature becomes uniform (*10*). To be successful, this approach requires severe temperature changes or the existence of a deep crack, and rotors that can safely withstand

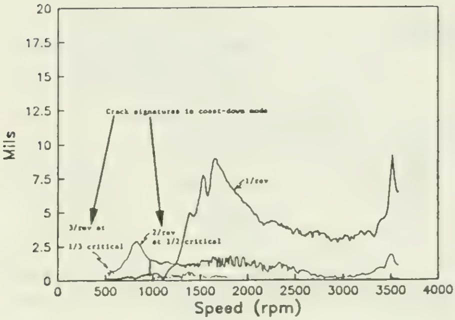


Figure 4 – Utility steam turbine rotor crack vibration data indicating 2/rev and 3/rev coast-down responses.

shock – a condition often promoting further cracking and distortion.

HISTOGRAM ANALYSIS

Under EPRI contract RP1862-2, GE developed an improved signature technique to detect shallow transverse cracks while machinery is fully operational (*11*). This technique applies to any rotor that bends – from its own weight, imbalance, or radial thrust loads – causing a crack to open and close. This crack movement or *breathing* causes rotor flexibility to change, a nonlinear effect that produces 1/rev and 3/rev signatures. Moreover, the opening of the crack introduces rotor flexibility asymmetry that is apparent in the 2/rev response.

The histogram technique reduces background vibration noise and eliminates the harmonics of existing normal machine operation. The averaged vibration data for an uncracked rotor become the normal operating baseline. New averages are collected periodically, subtracted from the baseline in the time-domain, and converted to the frequency-domain. The resulting differential harmonics – vector histogram harmonics – contain both amplitude and phase information. A third-order least-squares trend plot of the histogram harmonics is updated at each analysis period. This signal enhancement makes it possible to detect the initial appearance and steady increase in the 1/rev, 2/rev, and 3/rev harmonics.

The coast-down analysis is also performed using the histogram technique. A plot of the coast-down histogram harmonics versus speed will produce a 2/rev peak at one-half critical speed and/or a 3/rev peak at one-third critical speed if a crack exists. This procedure is estimated to improve sensitivity from 25% to 10-15% crack depth. Temperature-transient analysis sensitivity is similarly improved by the histogram technique. Further details on the histogram analysis are given in Reference 11.

Figure 5 shows spectral and histogram trend plots produced from a large-scale laboratory test where transverse cracks were grown in a 6-in. diameter shaft on a rotating fatigue machine. The steady-state data confirmed that the 2/rev histogram begins to increase with very shallow cracks of only 1-5% deep. All cracks up to about 15% deep indicated a dominate 2/rev asymmetry effect, although those near bearings or coupling respond less than those near the rotor midspan. Deeper midspan cracks alter rotor flexibility, producing a dominant 1/rev response.

All three crack diagnostic techniques – coast-down, temperature-transient, and histogram analyses – are incorporated in the monitor. Only the steady-state histogram analysis is used in the expert system, while the others are available for supplemental analysis. Since field crack data are limited to deep crack incidents, further experience with growing shallow cracks is required to fully establish the histogram sensitivity on operating machines.

AUTOMATED DIAGNOSTIC SYSTEM

Since rotor cracks are very infrequent, the Port Everglades plant personnel requested that the monitor diagnostics be extended to other more common faults. The diagnostic system software modules for the expert system, data collection, and signature analysis are briefly described.

GEN-X

This software package is a shell for the creation of rule-based expert systems. The knowledge base is a collection of if/then rules presented in tables or trees. The system considers through an inference engine, an efficient combination of forward and backward chaining logic, the available information and determines the probability of each fault. If information is missing, the software can still draw a logical conclusion from what is available.

FEAT (FEATURE EXTRACTION AND ANALYSIS TOOL)

In this module, numerical information is converted into statements such as *true, false,* and *cannot answer.* Thus, FEAT acts as a *go-between* from the database containing sensor data to the GEN-X expert system requiring textual or symbolic responses. Transformation algorithms are used to extract features from the data as a single numeric value. This can be as complicated as the

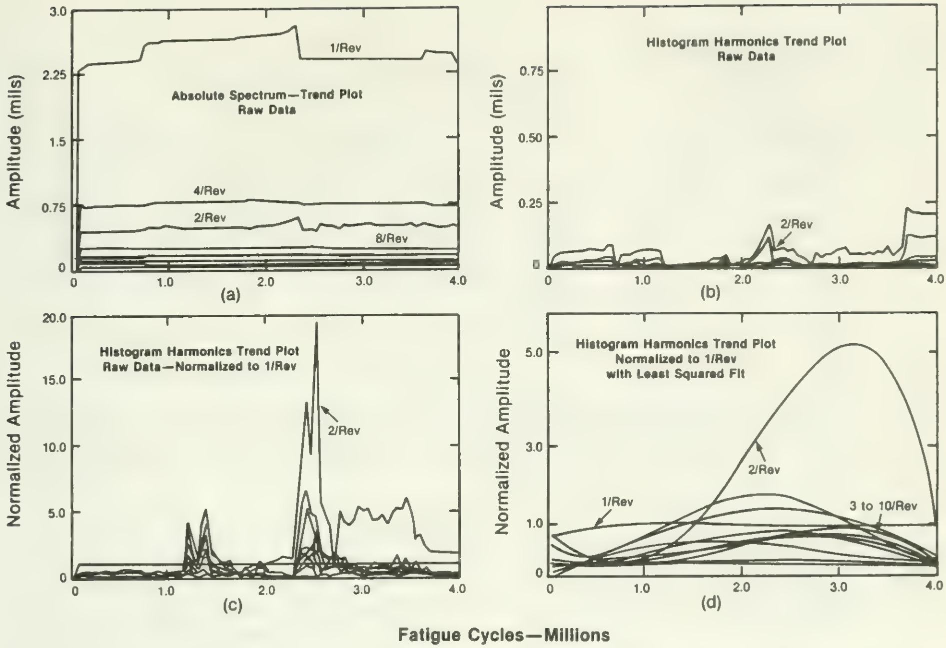


Figure 5–Rotor crack test plots indicate absolute and normalized histogram data. 2/rev histogram response begins at 1-5% and peaks at 15% of rotor depth.

maximum slope of a least-squares fit to the 2/rev vector histogram or as simple as the number of pad bearings in a turbine train. Each feature is then mapped against a fuzzy set or threshold, which defines the range for *true* and *false*, to derive the symbolic equivalent. Certain thresholds are machine and load dependent.

SIGNATURE ANALYSIS RULE BASE

This module contains about 100 rules and diagnostic strategies primarily derived from past crack diagnostics work and from vibration experts. Table I indicates the major fault types which can then be attributed to specific faults. For example, the system can determine if unbalance is due to blade breakage or erosion. Similarly, misalignment can be attributed to either the bearing or the coupling.

The rules take the form of if/then or condition/action statements. Conditions, such as an abnormal bearing metal temperature or an increasing 2/rev histogram established in the FEAT module, are used to determine whether a rule is true or false. A typical diagnostic rule checks whether a particular condition is true. If the condition is true, then a weighting factor–a measure of the condition's significance as a fault symptom–is applied. The summed weighting factors from each existing fault symptom are used to determine the likelihood of a suspected fault type. The rotor crack and misalignment diagnostic processes are presented below to illustrate the expert system-based monitor operation.

TRANSVERSE ROTOR CRACK DIAGNOSTICS

The monitoring system performs the three major vibration signature analysis procedures discussed earlier and outlined in Figure 6. This procedure must be considered tentative until sufficient field data are available for full validation. Since laboratory work had shown the steady-state histogram signature to be sensitive to very shallow cracks, only this technique is fully automated in the expert system computer.

TABLE I
EXPERT SYSTEM FAULT TYPES

Major Faults	Specific Faults					
Unbalance	Loss of Mass	Erosion	1st Stage Erosion	Stop Valve Bypass Failure	Bearing Wear	
Rub	Radial	Regular Radial	Carbonization Radial	Packing Rub	Axial Rub	
Bow	Water Induction	Thermal Sensitivity	Residual Bow			
Misalignment	Bearing	Bearing Vertical	Bearing Angular	Coupling	Parallel Coupling	Angular Coupling
Whirl	Oil	Steam	Resonance			
Mounting	Loose Bolts	Excessive Clearance	Bore Plug			
Rotor Crack	Transverse					

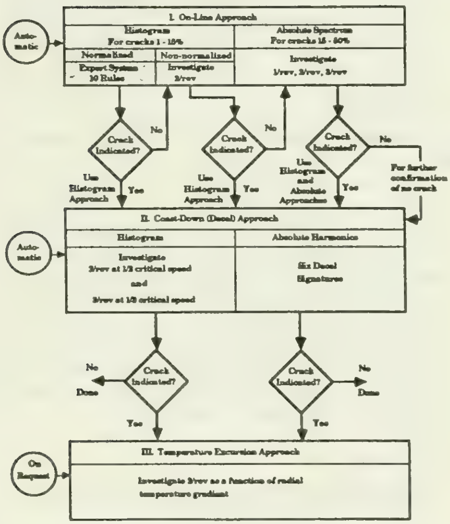


Figure 6 – Overall methodology for rotor crack detection using on-line (steady-state), coast-down, and temperature excursion/transient approaches.

ON-LINE STEADY-STATE ANALYSIS

The steady-state crack detection approach makes use of displacement data collected while the machine is at constant load and steam temperature. There are three methods of analysis. If a crack is indicated, the monitor automatically performs a coast-down analysis.

1. **Normalized Histogram.** 2/rev vector histogram data normalized to the 1/rev signal, and fitted to a third-order least-squares trend plot, are updated every 4 hours. The current numerical values for the histogram harmonics and curve fit slope are used to establish the conditions or facts shown in Table II as true or false. Symbolic facts are then used to answer the nine rules given in Table III. For instance, "Rule 1: If the 2/rev is dominant (Fact A), but not consistently (Fact B), then add weighting factor (W1). Weighting factors, ranging in value from 1 to 3, are summed from all rules found to be true." If this summed value exceeds a preset minimum value, such as 10, then the operator is alerted that a crack is indicated.
2. **Nonnormalized Histogram.** The 2/rev vector histogram is directly compared to other higher harmonics. If the 2/rev is found to be consistently larger, a coast-down analysis is performed.
3. **Changes in 1/rev, 2/rev, 3/rev Harmonics.** This method is effective for crack depths 15 to 50% of the rotor diameter. The following amplitude changes in the bearing displacement sensor data are used to trigger a coast-down analysis:
   - 0.5 mil 2/rev change
   - 0.5 mil 3/rev change
   - 1 mil 1/rev change

COAST-DOWN ANALYSIS

The coast-down analysis uses data collected during rundown of the machine from operating speed to turning

TABLE II
STEADY-STATE CRACK ANALYSES: SYMBOLIC FACTS

FACTS		TRUE	FALSE
A	IS THE 2/REV GREATER THAN THE 0 5/REV, 3/REV, 4/REV. 5/REV, ..., 10/REV?	T	
B	IS THE ANSWER TO QUESTION (A) TRUE AT LEAST 7 OUT OF THE LAST 10 TIMES?		F
C	IS THE 2/REV GREATER THAN 1.0 (1/REV)?	T	
D	IS THE ANSWER TO QUESTION (C) TRUE AT LEAST 7 OUT OF THE LAST 10 TIMES?	T	
E	IS THE 2/REV SLOPE GREATER THAN 0.5/REV,2/REV,3/REV,4/REV, ..., 10/REV SLOPES?		F
F	IS THE ANSWER TO QUESTION (E) TRUE AT LEAST 7 OUT OF THE LAST 10 TIMES?		F

TABLE III
STEADY-STATE CRACK ANALYSES: RULE BASE

RULES	IF TRUE	IF TRUE	IF FALSE	THEN ADD WEIGHT	FIRE	VALUE
1	A		B	W1	•	W1
2	B		A	W2		
3	A	B		W3		
4	C		D	W4		
5	D		C	W5		
6	C	D		W6	•	W6
7	E		F	W7		
8	F		E	W8		
9	E	F		W9		

The normalized histogram least-squared trend plot data are used to establish the facts outlined in Table II. These facts or conditions are then used to answer the rules in Table III. If the rule conditions are satisfied, then the rule is said to be *fired*. Each fired rule is assigned a weighting factor. Weighting factors are then summed as a measure of the crack fault determination certainty.

gear. Displacement data are collected every few seconds during the deceleration process. The coast-down analysis is used as confirmatory evidence of relatively large cracks.

1. **Decel Histogram Analysis.** Asynchronous data are collected every 3 seconds during turbine coast-down. Because the rotor is changing speeds, it is not possible to use time-domain averaging to suppress background noise. Therefore, the resultant histogram from two unaveraged responses contains roughly double the noise level. Data smoothing is employed to reduce this noise. This signature technique is referred to as the deceleration or decel histogram. The operator reviews the most current decel histogram for the 2/rev at one-half critical speed or 3/rev at one-third critical speed crack responses.
2. **Absolute Harmonics.** Waterfall plot monitor data are examined by the operator for the presence of the following six crack signatures:
   - 2/rev vibration at one-half critical speed
   - 2/rev vibration at one-third critical speed
   - Growing 2/rev signal at operating speed
   - Double critical speed signature
   - Reduction of critical speed
   - Persistence of resonance at critical speed

Further confirmatory evidence may be obtained for deep cracks within the steam path by manually initiating the temperature-transient procedure.

TEMPERATURE-TRANSIENT ANALYSIS

In this approach the 2/rev histogram data produced during a sudden 50-100 °F drop in steam temperature are collected by the monitor every 2 minutes over a 2-hour period. The induced radial temperature gradient will temporarily force open a crack on the outer rotor surface. A crack is indicated by an increasing, then decreasing, 2/rev signature.

MISALIGNMENT DIAGNOSTICS

Rotor cracks are sometimes misdiagnosed as misalignment because of the common 2/rev response. Table IV provides a comparison of rotor crack and misalignment signatures. To discriminate between the two fault types, the expert system must evaluate other sensor data. Misalignment diagnosis follows these steps:

1. Sensor data are collected once per hour and placed in a database. Bearing, coupling, and axial dc positions, bearing metal temperature, and displacement data are stored by time, load, and steam temperature.
2. The numerical-to-symbolic data conversion is outlined in Table V. Each fact or condition described is either true or false.
3. The symbolic facts are used to respond to rule base questions in Table VI. Screening rules determine probable major faults, followed by a general and specific fault analysis (see Fault Type, Table I). If dc position or bearing metal temperature is abnormal, the general and specific case for misalignment is investigated. Each rule found to be true is assigned a weighting factor proportional to its importance. A total weight for each investigated major fault is determined.

4. Major faults are ordered from highest to lowest nonzero total weight. The specific fault determination is listed along with the associated major fault.

Misalignment and transverse rotor crack faults are likely to be interconnected. For example, a shift in the turbine generator foundation may produce a misaligned rotor

TABLE IV
CAMPARISON OF ROTOR CRACK
AND MISALIGNMENT SIGNATURES

Symptoms	Crack Signature	Misalignment Signature
1. 2/rev	Yes (High)	Yes (Medium)
2. Rate of Change of 2/rev	Very High	Low ⟶ Zero
3. Rate of Change of 1/rev	Slow in Beginning; High for Deep Crack	Low ⟶ Zero
4. Direction of 2/rev	Radial	Axial
5. 2/rev Trend	Slowly Growing	Essentially Constant
6. Phase Shift	Not Necessary	Changes
7. DC Position	No Change	Changes
8. Bearing Metal Temperature	No Change	Changes
9. Coast-Down (Decel)	2/rev at 1/2 Critical	Should not be Affected
10. 2/rev at Motor	Yes - When Connected No - When Not Conn.	Not Affected
11. Probe Location	Most Probe Show	Mostly Local Phenomenon

* Crack vs. misalignment comparison rules not implemented in the prototype system

Trending of the 2/rev and 1/rev vibration changes are key to discriminating a crack from a misalignment fault. Frequent data collection during coast-down and steady-state is necessary to produce trend plots.

with a sufficiently large bending moment to initiate a fatigue crack. The expert system would evaluate the combined symptoms and identify both misalignment and rotor crack faults. Providing the crack continues to propagate, the rise in the 2/rev histogram harmonic detected by the monitor would alert the operator to the more serious condition. However, a shallow crack that grew over a short period of several hours and then was arrested would be difficult to detect confidently with current diagnostics. Detailed fault development experience on fully monitored machines, such as the Port Everglades installation, is required to validate the adequacy of the sensor data processing and diagnostic procedures.

UTILITY APPLICATION PROSPECTS

The automated expert system interpretation of sensor data holds promise to improve the effectiveness of utility periodic and continuous condition monitoring programs. Large amounts of data collected from periodic machinery surveillance programs using portable vibration spectral collectors and from continuous monitoring turbine supervisory instrumentation can be more efficiently screened and related to performance and maintenance data using expert system guidance. Since an expert system can readily supply routine fault analysis, vibration and equipment specialists will be able to focus on more unusual events.

Rotor crack diagnostic techniques are not widely understood since cracks rarely occur. Most cracks are not detected until repeated attempts to align and balance the rotor fail to eliminate high vibration. This approach has allowed steam turbine rotor cracks to grow to depths of 50% or more. An expert system-based monitor could greatly reduce this potentially catastrophic condition, particularly for older rotors that have grown brittle and

TABLE V
MISALIGNMENT FACTS

FACTS	MISALIGNMENT ANALYSES: SYMBOLIC FACTS			
		TRUE RANGE	TRUE	FALSE
A	ARE THERE ANY ABNORMAL D C . POSITIONS?	>9 mils	T	
B	ARE THERE ANY ABNORMAL BEARING METAL TEMPERATURES?	>15F		F
C	IS THE 1/REV PHASE STEADY?	<10deg/hr	T	
D	IS THE 2/REV PHASE CHANGING?	>20deg/hr	T	
E	IS THERE A SIGNIFICANT DIFFERENCE BETWEEN ADJACENT BEARINGS' METAL TEMP S?	>30F		F
F	IS THERE A SIGNIFICANT DIFFERENCE BETWEEN ADJACENT BEARINGS' ORBITS?	manual	T	
G	IS THERE A SIGNIFICANT DIFFERENCE BETWEEN ADJACENT BEARINGS' D C POSITIONS?	>9 mils	T	
H	ARE ANY COUPLING D C POSITIONS ABNORMAL?	>9 mils	T	
I	ARE ANY AXIAL METAL TEMPERATURES ABNORMAL?	>15F		F
J	ARE ANY AXIAL D C POSITIONS ABNORMAL?	>9 mils	F	
K	IS THE 2/REV VIBRATION COMPONENT ABNORMAL?	>0.8 mils	T	
L	DID THE RELATIVE D C POSITION OF ADJACENT COUPLING PROBES CHANGE?	> 9 mils	T	
M	DID THE RELATIVE PHASE OF ADJACENT COUPLING PROBES CHANGE?	>10 deg	T	
N	IS THE 1/REV VIBRATION COMPONENT ABNORMAL?	>1.8 mils		F
O	IS THE SUB-SYNCHRONOUS VIBRATION COMPONENT ABNORMAL?	>0.5 mils		F

Note: Representative values for the *true range* are given. Values, however, are machine and load dependent.

TABLE VI
MISALIGNMENT RULES

	MISALIGNMENT RULES	TRUE	FALSE	FIRE	VALUE
MAJOR FAULT SCREENING (PARTIAL LISTING)					
1	IF abnormal D C position THEN investigate MISALIGN	T		•	
2	IF abnormal bearing metal temperature THEN investigate MISALIGN		F		
GENERAL MISALIGNMENT					
3	IF 1/rev phase is steady and 2/rev phase changes THEN add W3	T		•	W 3
4	IF bearing metal temperature is abnormal and D.C. position is		F		
	abnormal THEN add W4				
5	IF there is a significant difference between adjacent bearings'	T		•	W 5
	metal temperatures or orbits or D.C position THEN add W5				
6	IF any coupling D.C. positions are abnormal THEN add W6	T		•	W 6
7	IF axial metal temperature is abnormal THEN add W7		F		
8	IF axial D.C. position is abnormal THEN add W8		F		
ANGULAR COUPLING MISALIGNMENT					
9	IF axial metal temperature is abnormal THEN add W9		F		
PARALLEL COUPLING MISALIGNMENT					
1 0	IF relative changes in D.C. position and phase occur between	T		•	W 1 0
	adjacent coupling probes occur THEN add W10				
VERTICAL BEARING MISALIGNMENT					
1 1	IF 1/rev and sub synchronous is abnormal THEN add W11		F		

The vibration, rotor position, and bearing metal temperature sensor data are used to establish the facts in Table V. Each fact is determined to be either true or false. Unavailable sensor data are treated as an unknown fact and dropped from the analysis. The rules given in Table VI are arranged to determine first the likely major faults and then to proceed with a more detailed analysis to confirm the fault type and its mechanical cause.

contain many ultrasonic testing indications. If a crack is detected early, the rotor may be salvaged by machining or weld repair.

FIELD TEST UPDATE AT FLORIDA POWER AND LIGHT (FPL) PORT EVERGLADES PLANT

In October 1986, a microprocessor-based vibration signature analysis monitor was installed on a steam turbine generator at Florida Power and Light Co., Port Everglades station. Initially the monitor consisted of a data aquisition system with built-in diagnostics for rotor crack. In 1988, the rotor crack diagnostics were reformulated and additional fault symptoms added to an interconnected expert system. Data collection, processing and numeric analyses are performed by an HP computer, while expert system diagnostics is performed by a Compaq 286. In 1989 the system is continuing to monitor the FPL steam turbine generator while it is on-line. Monitoring occurs during normal operations, as well as during coast-down of the turbine.

Vibration data (phase frequency and magnitude) are collected from ten probes located on the machine. Figure 7 indicates the location of the probes, as well as the turbine generator machinery arrangement. Data are entered into the database approximately every hour. Diagnostics can be performed manually by viewing on-line absolute harmonic data, on-line histogram harmonic data, and

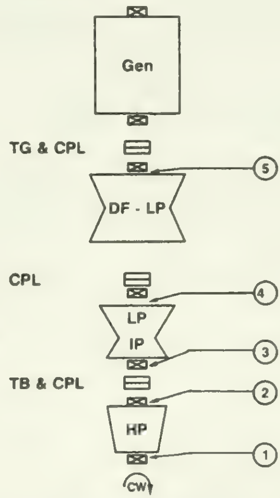


Figure 7 – Florida Power and Light turbine generator machinery arrangement and probe locations.

coast-down data or automatically by the automated expert system. Figures 8 through 15 are examples of vibration performance plots of the data collected during the month of March and April from the FPL monitoring system.

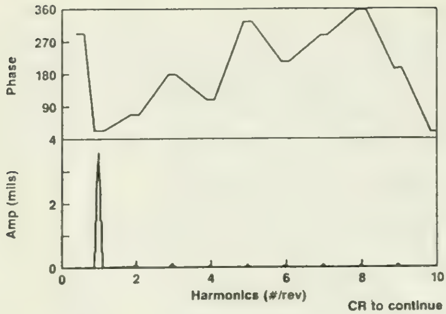


Figure 8 – Plot of on-line, absoute spectral harmonic data from FPL turbine generator on April 4, 1989.

ON-LINE ABSOLUTE HARMONIC ANALYSIS

Absolute spectral harmonics data are used to determine abnormal levels or trends in the magnitude and phase of the harmonics, which may indicate a crack or other flaw symptoms. Figure 8 shows a spectral harmonic plot, including phase and magnitude of subsynchronous and supersynchronous harmonics, from data gathered at probe 7. Signal averaging is performed to enhance the signal-to-noise ratio. Trended spectrum data are presented in two forms. One form is a plot of the magnitude of a harmonic versus time. Figure 9 is an example of this form, showing the trend of the 1/rev, 2/rev, and 3/rev harmonics during the month of March. A third-order least-squares curve has been fit to the data to provide smoothing. Trended spectrum data are also presented in the form of a Bode plot as shown in Figure 10. This form provides graphical display of amplitude and phase trend information.

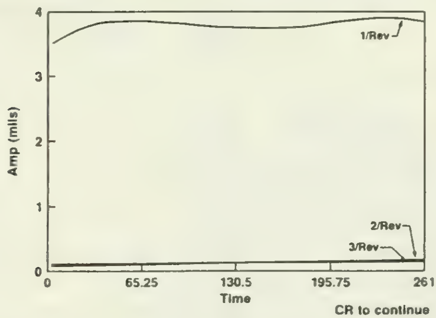


Figure 9 – Magnitude vs time trend plot of on-line, absolute spectral harmonic data from FPL turbine generator between March 2 and April 4, 1989. A third-order least-squares curve has been fit to the data.

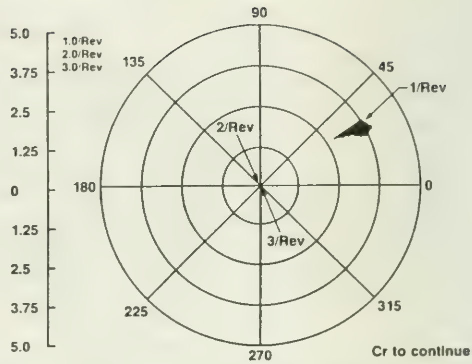


Figure 10 – Bode plot of on-line, absolute spectral harmonic data from FPL turbine generator between March 2nd and April 4, 1989.

ON-LINE HISTOGRAM ANALYSIS

Histogram data are used to show the change in the machine harmonics that is due to a change in the condition of the machine by vectorially subtracting a baseline signal from the current signal. A change in the 2/rev harmonic exceeding the change in the 1/rev harmonic is possibly an indication of a crack. Similarly, a change in the subsynchronous harmonic exceeding the change in the 1/rev harmonic is possibly an indication of whipping problems. Figure 11 shows a histogram plot. Comparison of the subsynchronous and 2/rev harmonics from this histogram with the same harmonics in the corresponding absolute spectral harmonics plot in Figure 8 shows how the histogram analysis enhances changes in the spectrum data. The activity in these two harmonics is being watched closely; however, until their level exceeds that of the 1/rev, there is no indication of a flaw. A third-order least-squares trend plot of the histogram harmonics is updated at each analysis period. Figure 12 shows a plot of a histogram trend for the 1/rev, 2/rev, and 3/rev harmonics.

COAST-DOWN ANALYSIS

Coast-down analysis is also performed by the monitoring system during rundown of the machine. When the rotor speed drops below a certain level, displacement data are collected every few seconds from probes 1, 5, 8, and 10. The data can be viewed in either absolute or histogram form. A crack would be indicated by a peak in the 2/rev at one-half critical speed and a peak in the 3/rev at one-third critical speed. Figure 13 shows a decel

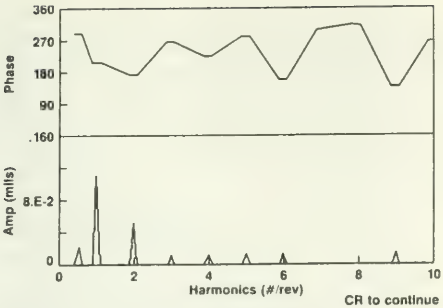


Figure 11–Plot of on-line histogram data from FPL turbine generator on April 4, 1989.

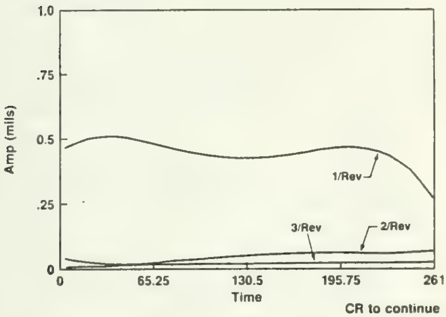


Figure 12–Trend plot of on-line histogram data from FPL turbine generator between March 2 and April 4, 1989. A least-squares curve has been fit to the data.

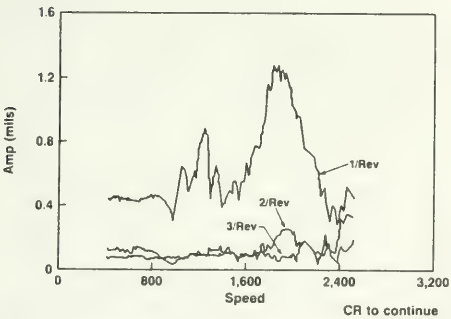


Figure 13–Decel histogram plot of coast-down data from FPL turbine generator on March 31, 1989.

histogram. The absence of the previously mentioned symptoms indicates that no crack is present. The high level of noise occurs because the changing rotor speed does not allow for signal averaging.

AUTOMATED EXPERT SYSTEM-BASED DIAGNOSTICS

The expert system receives data each day and performs diagnostics automatically, while the turbine generator is on-line. The results of the diagnostics are displayed on the screen in either of the forms shown in Figures 14 and 15. The flaws that can be detected include unbalance, bowing, rubbing, misalignment, instabilities, mounting problems, and rotor cracks. If the expert system suspects a flaw but is unable to detect it with absolute certainty, it will alert the user to the possibility of the flaw's existence. In addition to the automatic mode, the expert system can perform in an interactive mode, as well as a batch mode.

To date, the monitoring system has been operating continuously in an automatic mode essentially flawlessly. It has yielded valuable vibration information as shown above. No flaws of appreciable size have yet been indicated by the expert system or by human review of trend and coast-down plots of spectrum and histogram harmonics.

Date	Time	Unbalance	Rubbing	Bowing	Misalign	Whirl	Mounting	Crack
04/04/89	756	No	-	No	-	No	-	No
04/04/89	556	No	-	No	-	No	-	No
04/04/89	456	No	-	No	-	No	-	No
04/04/89	355	No	-	No	-	No	-	No
04/04/89	255	No	-	No	-	No	-	No
04/04/89	155	No	-	No	-	No	-	No
04/04/89	55	No	-	No	-	No	-	No

Figure 14–History of results of expert system diagnostics.

Florida Power and Light
Unit #1

Based on information between: 9/6/88, 851 and 4/3/89, 1154

– NON-CRACK FLAWS –
No non-crack flaws indicated

– CRACK FLAW –
Crack is not indicated

Figure 15–Results of expert system diagnostics.

CONCLUDING REMARKS

An expert system approach has been described that automates the diagnosis of a transverse steam turbine rotor crack and other common vibration fault signatures. To use rule-based knowledge representation, a method was developed for transforming quantitative, numerical sensor data from the monitoring system to the qualitative, symbolic responses required for an expert system. The expert system evaluates multiple-fault symptoms, including vibration response, bearing temperature, and shaft position, obtained from machine sensors monitored by a dedicated data acquisition system, using weighting factors.

The expert system-based, on-line vibration signature analysis monitor is intended to diagnose transverse rotor crack and other fault symptoms produced during steady-state, coast-down, and steam temperature transients without operator review. Further development is still required to meet this goal fully. Field testing on an operating utility steam turbine is under way to validate the expert steady-state vibration system and histogram technique. Laboratory results indicate a crack depth detection threshold of 1-5% of the rotor diameter. This prototype expert system assists in discriminating a crack from other faults, such as misalignment or unbalance, which produce similar symptoms. The expert system knowledge base can be easily modified to include supplemental and confirmatory rotor crack signatures and other steam turbine fault symptoms necessary to increase the confidence of diagnosis.

REFERENCES

1. EPRI Seminar on Expert Systems Applications in Power Plants, Boston, MA, May 27-29, 1987. To be published by EPRI.
2. Roberts, A., "Vibration Signature Analysis as a Means of On-Line Nondestructive Examination of Turbine Generators," *EPRI Nondestructive Evaluation of Turbines and Generators Conference Proceedings* WS-80-133, July 1981.
3. Gonzalez, A. J., Osborne, R. L., Kemper, C. T., Lowenfeld, S., "On-Line Diagnosis of Turbine Generators Using Artificial Intelligence," *EPRI Incipient Failure Detection in Power Plants Conference Proceedings* CS-4748, Orlando, FL, October 10-14, 1984.
4. Gopalakrishnan, S., "Boiler Feedpump Expert System," presented at EPRI Power Plant Pumps Symposium, New Orleans, LA, March 10-12, 1987.
5. Vinson, J., and Perez, W., "Turbomac: An Expert Machinery Diagnostic System," presented at the EPRI Seminar on Expert System Applications in Power Plants, Boston, MA, May 27-28, 1987. To be published by EPRI.
6. Mayes, I. W., "Crack Propagation in Rotating Shafts," ASME Paper 77-DET-164, presented at the Design Engineering Technical Conference, Chicago, IL, September 26-30, 1977.
7. Henry, T. A., and Okah-Avae, B. E., "Vibrations in Cracked Shafts," Conference on Vibrations in Rotating Machinery, University of Cambridge, September 1976.
8. Mayes, I. W., and Davies, W. G. R., "The Vibration Behavior of a Rotating Shaft System Containing a Transverse Crack," Conference on Vibrations in Rotating Machinery, University of Cambridge, September 1976.
9. Gash, R., "Dynamic Behavior of a Simple Rotor with a Cross-Sectional Crack," Conference on Vibrations in Rotating Machinery, University of Cambridge, September 1976.
10. General Electric, Technical Information Letter TIL-727-2, March 18, 1974, and TIL-750-3, October 11, 1974.
11. Imam, I., Scheibel, J. R., Azzaro, S. H., and Bankert, R. J., "Development of an On-Line Rotor Crack Detection and Monitoring System," *1987 ASME Design Technology Conference – 11th Biennial Conference of Mechanical Vibration and Noise Proceedings*, Boston, MA, September, 1987.

Development, Customization, Installation of PERFEXS: A Power Plant Performance Diagnostics Expert System

F. FRANCO, M. ORIATI, and A. SERVENTI
ANSALDO
Development Engineering Department
Via dei Pescatori 35
16128 Genova, Italy

P. RIBALDONE and V. VELLINI
Azienda Energetica Municipale
Via Bertola 48
10122 Torino, Italy

ABSTRACT

This paper describes criteria, problems and results related to the phases of development, customization and installation of the expert system PERFEXS, developed at ANSALDO Development Engineering Department and installed for operation testing at Moncalieri, a 136 MWe power plant owned by AEM, the Municipal Utility of Torino, Italy.

PERFEXS (PERformance EXpert System) is an on-line expert system for power plant performance diagnostics: it provides continuous plant performance monitoring, detection of deviations from optimum performance, diagnosis of the causes of such deviations, suggestions of possible corrective actions. PERFEXS is delivered on a 386 Personal Computer.

A short description of PERFEXS characteristics and of the Moncalieri plant and process configuration is provided in the paper.
The process of adapting PERFEXS, intended as a general diagnostic tool, to the specific application is illustrated in some detail. This means incorporating specific plant physical and process characteristics, specific utility operational needs and specific user requirements for man-machine interface into the general tool.
Considerations are also reported on the problems involved by the insertion of the system into the existing plant instrumentation, monitoring and control system, as well as into a well established operational behavior.

1. INTRODUCTION

PERFEXS (PERformance EXpert System) is an on-line expert system for power plant performance diagnostics developed at the Development Engineering Department of ANSALDO S.p.A. (Ref. 1).

ANSALDO is a major Italian company in power plant engineering and manufacturing. Since 1986 it is committing resources to the development of industrial applications of expert systems.

AEM (Azienda Energetica Municipale) is the Municipal Utility of Torino, Italy. It operates several thermal and hydro power plants, serving the local area and connected to the national network. Among these, Moncalieri is a 207 MWe thermal power plant, oil and gas fuelled.

An application of PERFEXS to the Moncalieri 136 MWe second unit, that is initially limited to diagnosing the performance of the condenser system, has been developed and is now being installed at the site. Operation testing will take place thru 1989.

This paper covers the following main subjects:

- functions and main architecture characteristics of PERFEXS, as a performance diagnostics general environment for specific applications in this paradigm
- Moncalieri plant and process characteristics, operational environment and plant management needs
- customization of PERFEXS to Moncalieri plant: incorporation into the system of specific plant parameters and process characteristics; satisfaction of specific user needs and incorporation of plant operating staff experience; presentation of information according to user's requirements
- installation of PERFEXS in a plant environment not oriented to host informatic applications

2. PERFEXS DESCRIPTION

PERFEXS (*) is a support to the plant staff in improving fuel consumption efficiency. It is able to monitor on-line the plant performance from the heat consumption standpoint, to detect deviations from the expected (optimum) performance, to diagnose the cause of such deviations, to suggest possible corrective actions in order to eliminate malfunctions. On-line detection of excessive fuel consumption and prompt identification of causes and remedies are expected to have a great potential for significant savings in plant operation costs.

2.1. General system architecture

The plant representation adopted in PERFEXS for the purpose of performing diagnostic functions schematizes the plant in five main subsystems: boiler, turbogenerator, condenser, water regeneration, electrical auxiliaries.

For each of these a set of index variables is defined, i.e. the process variables representative of the subsystem performance. The NOMINAL (optimum) values of index variables depend on actual plant operating conditions (independent variables), and are in general dynamically calculated by PERFEXS making use of analytical models and on the basis of the current operating conditions (power level, cooling water temperature,....etc).

The MEASURED values are taken from the field and compared to NOMINAL values. This is performed periodically for each subsystem. If the differences between the two sets exceed an established tolerance, the diagnostic process is triggered for the involved subsystem.

This process identifies the causes of deviation and suggests the way to remove them.

The diagnostic process is performed by a general inference engine that operates on a knowledge base specific to each subsystem.

Field data are normally taken from the data base of the existing plant supervision system. In the case of Moncalieri a dedicated data acquisition system was installed.

A schematic of PERFEXS architecture and of its typical plant integration is reported in Fig. 1.

(*) Part of the development work for PERFEXS was performed by ANSALDO in the frame of ESPRIT Project 820 sponsored by EEC. Other partners of the project were: CISE (Italy), Aerospatiale, FRAMENTEC and CAP SOGETI (France), F.L. Smidth (Denmark), Heriot-Watt University of Edinburgh (United Kingdom).

2.2 PERFEXS design criteria

The necessary compromise between user and developer expectations (the first one requires a system tailored on his specific application, the second one requires to implement a standard system for different applications) represents the equilibrium point of the cost-profit ratio of an informatic product.

This led ANSALDO to implement a diagnostics oriented general system capable of being easily customized into specific diagnostic delivery systems.

In the case of PERFEXS the above mentioned requirements led system and software engineers to identify two classes of design criteria to be complied with.

The first class includes architectural features:

- a modular architecture of basic functions of the system
- a capability to integrate the system within the plant information and automation system
- a man-machine interface homogeneous with the existing environment and user habits.

The second class includes structural features:

- a knowledge organization and an inference engine structure which are general with respect to plant or plant subsystem
- a knowledge acquisition tool which exploits the general knowledge organization allowing the knowledge engineer to personalize the knowledge base for the specific application in a simple way
- several dedicated development tools which allow personalizing the general modular system functions (data acquisition, data management, man-machine interface, etc).

Meeting such criteria allows for a suitable case-by-case satisfaction of specific requirements of the system end user, while at the same time, thanks to the built-in personalization tools, minimizes the related costs.

In summary, the above mentioned features allow considering PERFEXS as an environment specialized in the specific consultation paradigm: PLANT DIAGNOSTICS.

2.3 PERFEXS Knowledge Base

The Knowledge Base (KB) is made of KNOWLEDGE CATEGORIES, each of which is a set of empty frames (object, object attributes, attribute values) that is filled, by means of a knowledge acquisition tool, with the specific knowledge pertaining to the plant subject of the specific application.

Thus, the category CORRELATIONS is a structure allowing the comparison of the nominal and measured values of the plant variables. An attribute of a CORRELATION is the SATISFACTION DEGREE, whose value may be one of the set <GREATER THAN>, <EQUAL>, <LESS THAN>, etc., depending on the result of the comparison.

The category of RULES is a structure which contains the association between a CORRELATION (with a specific SATISFACTION DEGREE) and a set of OBJECTS which can (or can not) be involved in the malfunction to be detected.

OBJECTS is a category that generally contains parts of plant equipment or their behavior versus a process variable. The category of DEPENDENCES is a structure defining a hierarchical link among OBJECTS, corresponding to different detail levels in plant representation.

The category of CAUSALITIES defines the possible malfunctions, faults or degraded perfomances of each OBJECT.

A member of the category ACTIONS is associated to each CAUSALITY, defining suggestions for the user (actions, verifications, etc.).

As it can be seen, the KB structure organizes quite in a general way the conceptual categories used in a diagnostic process (symptoms, association of symptoms with parts involved, causes, remedies). However, the content of such structure is plant and plant subsystem dependent (even though one may expect that not all of the information pertaining to, say, the boiler would change from a specific application to another).

2.4 PERFEXS Diagnostic Inference

The Inference Engine of PERFEXS exploits the specific contents of the KB categories to reach the goal of identifying the actual cause of malfunction. The process is schematically illustrated below.

PERFEXS checks the CORRELATIONS involving the index variables of a subsystem and, in case of mismatch between measured and nominal values (SATISFACTION DEGREE different from <EQUAL>) starts the diagnostic process.

A mismatch in a CORRELATION involves an associated RULE that recalls a set of OBJECTS potentially involved in the malfunction.

Further CORRELATIONS are checked and related sets of OBJECTS identified. A part of Inference Engine called Result Management Structure is in charge of dynamically classifying the OBJECTS.

The general strategy of OBJECTS management consists in distributing them into three dynamic classes:

- IMPLICATIONS: set of OBJECTS which are possible cause of malfunction (still to be verified)
- EXCLUSIONS: set of OBJECTS that are not possible cause of malfunction
- CONCLUSIONS: OBJECTS confirmed as possible cause of malfunctions

At the end of the process, if the diagnosis is successful, the class IMPLICATIONS is empty and the class CONCLUSIONS contains the culprit(s) of the malfunction.

The path for progressive checking of CORRELATIONS is governed by a part of the Inference Engine called Search Optimization Structure, which optimizes the choice of subsequent steps based on results of the previous steps.

A "single fault" assumption is at the base of the diagnostic process. The process may end with the sure identification of the cause of the malfunction, or with the unability to decide among a set of possible causes (which are shown to the user anyway). It may also end with a conflict situation, i.e. the determination of contradictory conclusions during the diagnosis (this may typycally depend on measurement failures or multiple faults in the plant): in this case the user is alerted of the conflict and the information available is supplied.

In summary, the Inference Engine is a set of general mechanisms for managing the knowledge as organized in the described KB; hence, it is completely independent of the plant and the plant subsystem.

3. MONCALIERI PLANT PROCESS CHARACTERISTICS

The Moncalieri 136 MWe, gas and oil fuelled second unit was put into service in 1967.

The application of PERFEXS described in this paper regards, at the moment, the condenser subsystem. A short description of the plant condenser and associated systems follows. A scheme of the condenser subsystem (as presented by PERFEXS) is shown in Fig. 2.

Steam discharged from the turbine is condensed in a surface condenser, and is extracted via a hotwell by two parallel 100% pumps (one stand-by). Uncondensable gases are extracted from the condenser via two vacuum pumps, each sized for 50 % charge.

Condenser cooling is effected by pumping channel water and feeding it to the condenser tube side via the cooling circuit shown in Fig. 2. Water intake is by two parallel 50% systems, each including a fixed grid, a rotating grid and a pump. The condenser is a two side-two passage type. Air extraction is provided in the condenser water boxes via vacuum pumps. Two separate discharge lines are provided for water return to the channel. Several cooling circuit configurations are possible:

- two-pumps/two-condenser-branches it is the normal operating configuration at high plant load
- one-pump/one-condenser-branch: it can be a low plant load configuration, or a "malfunction" configuration
- one-pump/two-condenser-branches: it can be a "malfunction" configuration, or a configuration selected for minimization of fuel consumption in particular cases (e.g. very low water temperature)
- two-pump/one-condenser-branch: it can be a "malfunction" configuration.

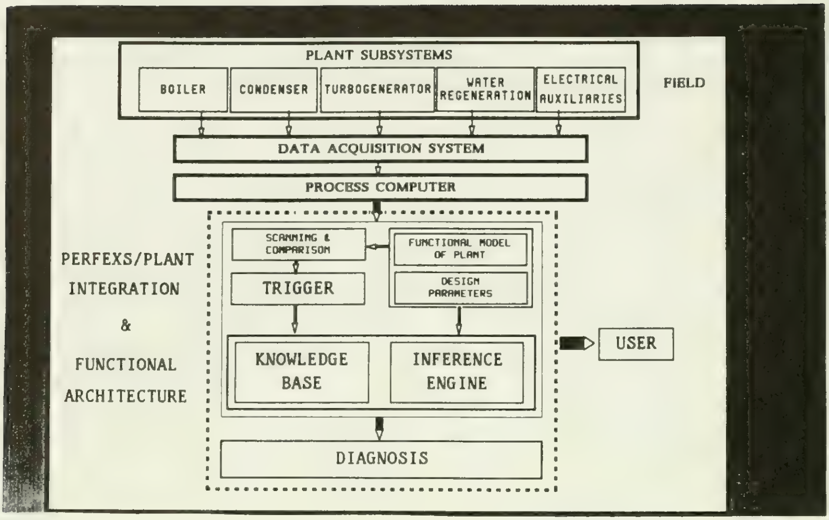


Fig. 1 - PERFEXS: Plant integration and functional architecture

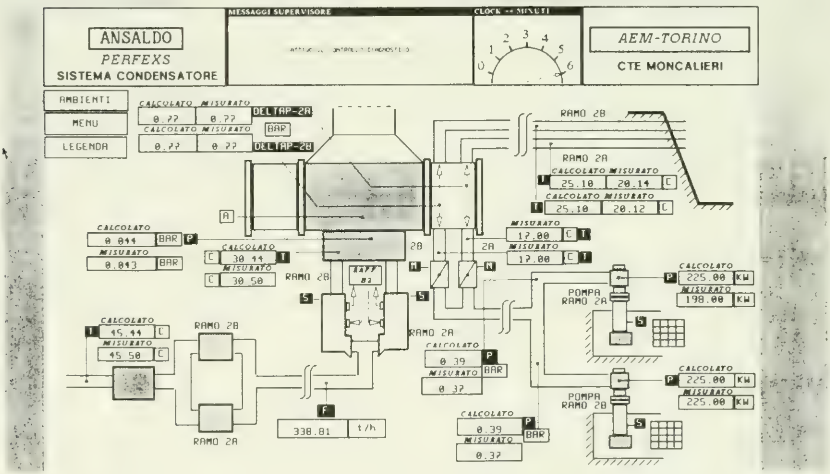


Fig. 2 - PERFEXS user interface environments: example of plant schematics

Performance of the condenser has a strong impact on plant overall performance, i.e. on fuel consumption. As a rough example, it can be estimated that a continuos loss of vacuum of 500 Pa (0.15 inHg) in the Moncalieri plant would cause an increase of fuel consumption of about 5000 m3/day of natural gas corresponding to a loss of performance of about 13 kCal/kWh (50 BTU/kWh).

Plant personnel is obviously well aware of this economic importance of the condenser system, and this is reflected into the plant manegement procedures and practice. In particular, condenser performance is reported periodically in terms of vacuum, water temperatures and water pressure drops, and compared to expected performance. Main comparision parameters are the cooling water pressure drops across the tube bundle, which are indicative of the condenser hydraulic behaviour.

Corrective actions, in addition to maintenance and repairs when necessary, mainly refer to the condenser tubes cleaning system (the taprogge system), which can recover the most likely type of malfunction, i. e. tube fouling. Taprogge system operation is recorded both in terms of time and of operational parameters.

The condenser diagnostic system is designed to incorporate all the functions of condenser performance control actually performed by hand, to add diagnostic capability, to provide operator guidance for corrective actions. End users of the system are the power plant manager, the maintenance engineer, the staff supervisor. The control room operators should have easy access to the system, with reference to all that information that may impact conduction parameters, such as configuration optimization, taprogge operation etc.

The condenser diagnostic system uses the following set of plant measures:

- condenser pressure
- condenser hotwell temperature
- 2 cooling water pumps power
- 2 cooling water pumps outlet pressure
- 2 branches cooling water inlet temperature
- 2 branches cooling water outlet temperature
- 2 condenser tube bundle pressure drop
- condensed steam flow rate
- first low pressure heater inlet temperature

4. DESCRIPTION OF THE CUSTOMIZATION PHASE

The aim of the PERFEXS customization phase was to integrate into the system the specific knowledge of the Moncalieri power plant condenser (measures, design parameters, operating configuration, etc.), as well as to modify or develop the system functions (mainly operator interface, reporting capabilities, etc.) as necessary to meet the requirements expressed by the system end user.

A side objective of this work was to verify the actual capability of the basic structures of the general system, as implemented in the development phase, to be specialized to the specific case flexibly and with a reasonable effort.

What follows outlines the problems tackled and the solutions adopted, in the cases considered of general interest in applying Knowledge Based Systems in the real context of a power plant. Three main aspects can be identified:

- Personalization of the knowledge base
- Transfer from development HW/SW environment to delivery environment
- Man-machine interface functions

4.1 Personalization of the knowledge base

The specific application activity started from an in-house demonstration case available at ANSALDO, in which inference structures and knowledge base organization were those outlined in paragraph 2, whereas the specific elements of knowledge were supplied by the design data of a reference condenser system.

The first step was a joint analysis (Moncalieri plant staff and ANSALDO system developers) of the configuration of the equipments involved in the performance diagnostic system and of the necessary set of design and operation data to be used as input to PERFEXS.
Two significant cases of personalization are described in the following.

The necessity came out to reconfigure the condenser system schematics and models for the application case: the main difference with respect to the reference model was a different physical configuration and different equipment set-up in the water side of the condenser, essentially consisting of a branching of the cooling water circuit into two parallel branches (instead of the single pipe of the reference model).

The hydraulic behavior of two parallel branches is significantly more complex to represent than a single pipe circuit, so that the analysis of symptoms related to the cooling circuit was not adequate, as it was performed in the initial model, to correctly manage the variety of occurrences and fault configurations actually possible in the real plant.

Therefore, it was necessary to develop, with the support of hydraulic engineering experts, some semi-quantitative mathematical models capable of representing the actual condenser system and suitable to be incorporated into the PERFEXS KB. After this modification, the diagnostic process was re-tested in laboratory, and the results were verified against the plant operating staff experience.

The case discussed above indicates that incorporating actual plant situations may require not only the support of plant operating experts, but also some iteration with the equipment designers.

Such iterations should hopefuly be less and less necessary, as experience is gained with several applications. It should be noted that, even though the engineering problem was not quite trivial, the solution could be incorporated into the KB without any change to its logic organization and general structure.

Another modification suggested by the analysis of operational needs was the implementation of an additional user interface allowing the operator to input to the system the date and time of operation of the Taprogge cleaning system; this information is stored by PERFEXS and is exploited at the KB level.
This allowed at least two improvements of the system: first, to be able to discriminate among different faults, thanks to the availability of an additional information; second, the capability to maintain a history log of the operations of the Taprogge equipment, that is considered useful for plant documentation purposes.

The latter described is an example of modification related to an increment of information and system capabilities connected with particular needs.

Other activities regarding KB personalization were more routinely plant information insertion into the system and are not discussed here in detail.

Some considerations may be drawn, in general, from this experience:

- As it was expected, the necessity was confirmed of an accurate analysis to be performed jointly among the system developers and users of the application from the early stage of the customization phase
- The opportunity was evidenced to formalize the interviews with domain experts in suitable formats as close as possible to the final informatic structure to be filled, in order to make the personalization process the easiest and most consistent possible.
- The desirability was clear, as an indication for the future, to implement an extremely user-friendly knowledge aquisition tool to be directly used by the domain expert.
- The modularity and generality carachteristics of the inference structures and of the knowledge base organization were confirmed by the practical use.

4.2 Transfer from development to delivery environment

The development environment for PERFEXS is KEE on LISP machine SYMBOLICS 3640, whereas the delivery environment is KEE version for PC 386 in the Operating System UNIX System V on PC COMPAQ 386 equipped with 13 Mb RAM and a 300 Mb hard disk.

For the Moncalieri application a dedicated Data Acquisition System (DAS) has been provided: this is a cabinet of the Distributed Control System ADAMS, designed and produced by ANSALDO. Also new instrumentation for taking field measurement to be input to the DAS has been installed and connected to the DAS. This instrumentation will also be used in the future for other plant supervision and control functions.

The DAS cabinet is equipped with the suitable input boards for signal acquisition from sensors in field and intelligent boards for data elaboration and transmission. The DAS cabinet is located in the equipment room, whereas the PC and its peripheral devices are located in control room at the upper floor; they are connected through a 9600 bps serial link for exchanging data and controls.

During the transfer from development to delivery environment two major aspects have been recognized, on which effort must be dedicated, with the objective to implement knowledge based products with industrial characteristics.

The first aspect is related to the changing status of the hardware and software base products market for expert system development. The continuos and fast changing and upgrading of products the market offers is well known for "conventional" HW and SW: this tendency is even more important in the case of the relatively young market of the Artificial Intelligence products.

In the experience during PERFEXS project, both with regards to HW devices and SW tools and environment, significant time and resources had to be devoted to updating and integrating base HW and SW tools to make them compatible with the overall system context.

Because industrial applications need generally relatively long development time and have even longer operational life and, more, they have in general to be integrated as functional modules into plant systems, the necessity is clear of selecting as standard as possible HW/SW base tools, even at the expense of some special or enhanced performances offered by new products in the market (even though this should not mean freezing the improvement and further development work). This is a condition for not making a one-of-a-kind case of each application, and for providing a well-established and reliable industrial product.

The second aspect is related to the integration of PERFEXS with plant systems.

The Moncalieri application is peculiar in this regard, because no automation functions based on computer systems were present in the plant. In this case, a dedicated DAS was developed and implemented, thus including in PERFEXS the data management tasks that would normally be performed by the plant information system and would be shared with other functional applications (based or not on expert system technology).

The DAS, however, is not specific to PERFEXS and can be shared with any additional function that might be required in the future.

A consideration may be that the convenience (in terms of installation costs vs. the entire system cost) of applications like this one, requiring additional instrumentation and data acquisition equipment, is, case-by-case, a function of what is already available at the plant and how much sharing of the additional equipment is possible among different functions.

More in general, the modular structure of PERFEXS is conceived for easily connecting via local network or serial link with plant automation/supervision system, with the exploitation of the existing instrumentation, data base and data management functions. This makes stronger the previously discussed incentive to standardization of HW/SW tools, in order to minimize integration problems.

4.3 Man-machine interface

PERFEXS has a dedicated user interface consisting of user selectable graphic video pages. In its design the leading criteria have been to maximize friendlyness and minimize the intervention of the user for requiring to the system the dynamically available information.

A hierarchical an d modular structure of the interface has been adopted. This is based on a set of "environments" that the user can access simply by clicking on a menu: each environment corresponds to a specific interface function which can have, if necessary, sub-environments also accesible by menu.

An INFORMATION environment is provided to inform the operator about the available enviroments and their access.

This structure allows for an easy case-by-case personalization and integration of the interface functions: the quality of the man-machine interface can strongly condition the operating success of any information system, therefore a lot of effort has been dedicated to this aspect during the development and personalization phase. On this point it is expected to have an important user feed-back from the operation phase.

The following are examples of some of the interface environments actually provided by PERFEXS:

- PLANT SCHEMATIC which is also the default environment: it shows a synoptic of the plant area interested in the performance diagnosis (fig. 2), also containing the values of relevant variables, both the measured value and the actual reference value, on-line updated with data coming from the field.

- DIAGNOSIS is the output environment automatically shown when a malfunction condition occurs. It has some sub-environments which can be accessed on operator request, e.g. PERFORMANCE which gives a quantification of the actual loss of performance of the plant in terms of kCal/kWh, and SUGGESTED ACTIONS which contains suitable information for the operator on "how to repair" the actual diagnosed fault.

- DIAGNOSIS EXPLANATION which allows the user to access step by step the diagnostic process followed by PERFEXS in the actual symptom consultation, once the diagnosis goal is reached. The output is based upon the structures in the knowledge base (CORRELATIONS, RULES, CAUSALITIES, etc.), but is presented in a non-coded "easy to read" format.

- HISTORY which shows the behaviour in the time of the plant measures in numerical or graphical form.

- TAPROGGE OPERATION INPUT which allows the operator to input and consult the operations of this equipment, etc.

Other environments are provided for configuring and tuning the system, e.g. DESIGN PARAMETERS SET-UP, CORRELATION THRESHOLD TUNING, MEASURE BY-PASS, REPORT TIMING, etc. , which generally should be accessed only at system installation or in rare cases like a measure out of order, changing of instruments, etc., but which allow the user to simply readjust the system on the basis of the actual operating requirements.

Fig. 3 illustrates the schematic general structure of PERFEXS vs. user and field interface and figs. 4 - 8 illustrate some examples of output environments.

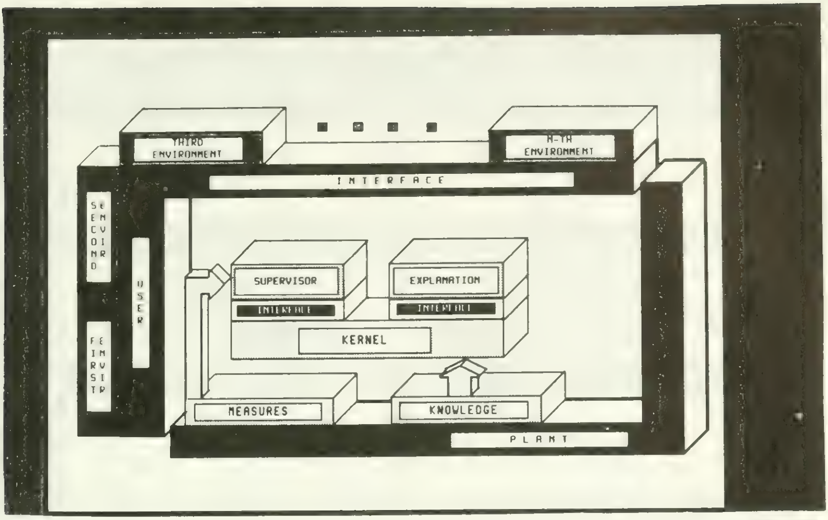


Fig. 3 - PERFEXS user and field interface modular structure

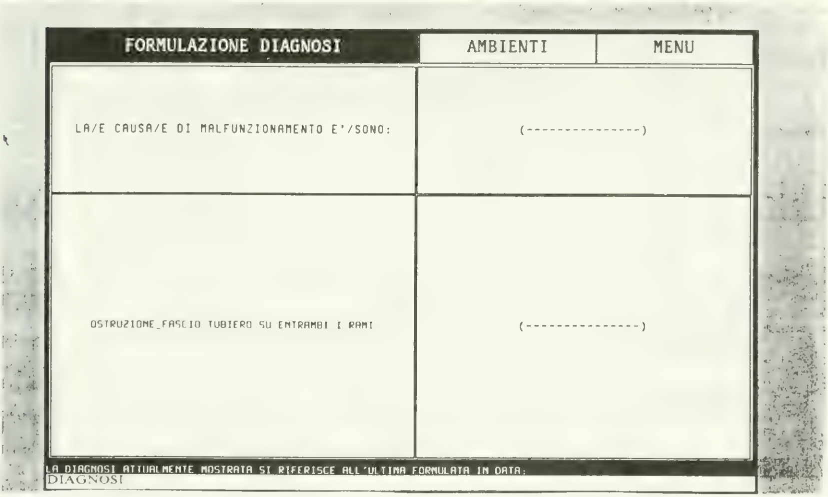


Fig. 4 - Environment DIAGNOSIS: example of fault detection

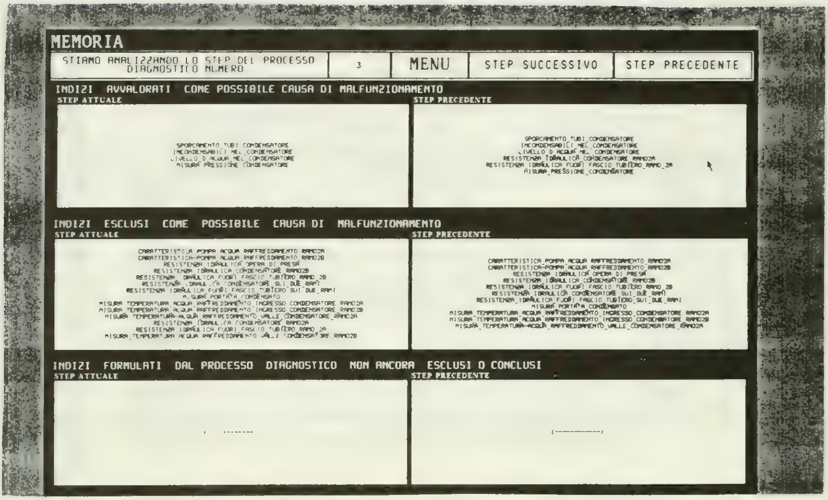


Fig. 5 - Environment DIAGNOSIS EXPLANATION: example of a step

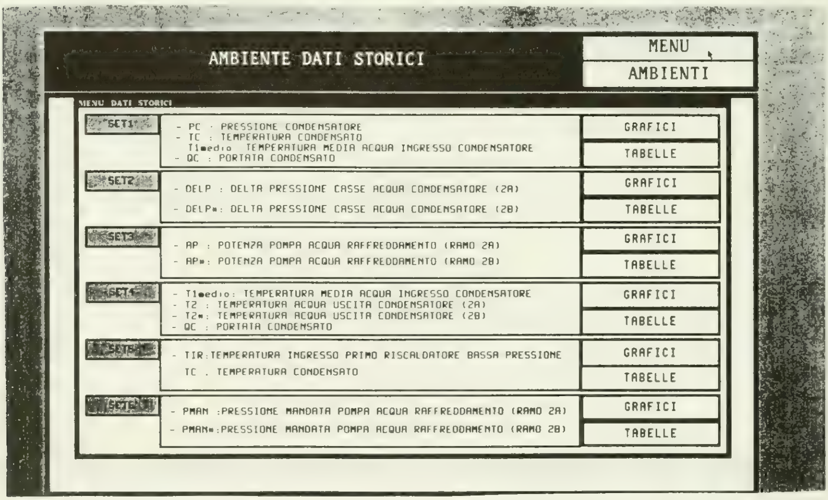


Fig. 6 - Environment HISTORY: user's menu

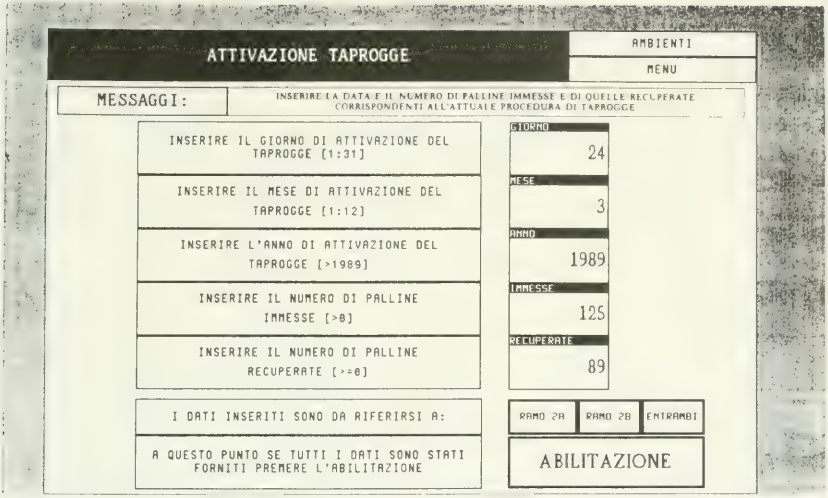


Fig. 7 - TAPROGGE OPERATION INPUT environment

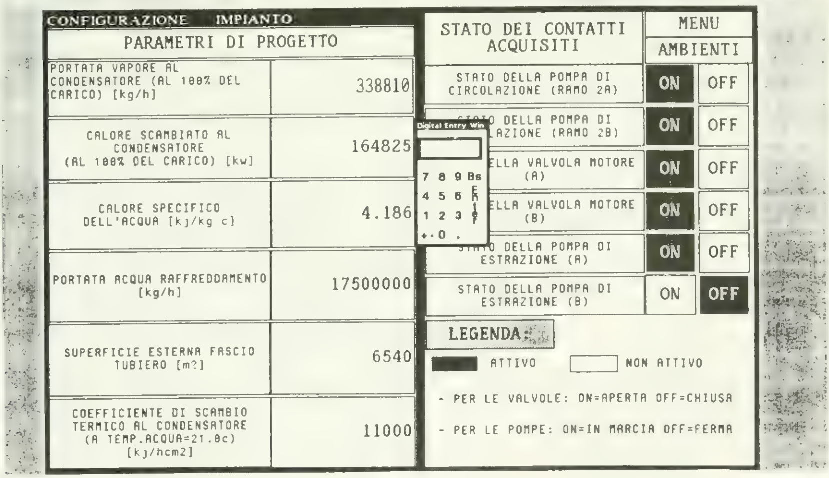


Fig. 8 - DESIGN PARAMETER SET-UP environment

5. CONCLUSIONS

As mentioned in para. 1, the PERFEXS application to Moncalieri is being installed at the plant at the time this paper is written. Immediately after, a period of site testing for system calibration will take place both with the plant at power and (in mid-summer 89) during plant shutdown. Operation testing will take place from September 1989 thru the end of the year.

The main considerations coming, as of today, from this application have already been elaborated in previous paragraphs. Summary conclusions are reported here:

- It was confirmed during the customization phase that the capabilities provided by the Artificial Intelligence techniques allow for significant advantages, in terms of flexibility and easiness of implementation, in adapting a general tool to the specific application needs and incorporating the specific user knowledge as an integral part of the system.

- A confirmation is now expected from the field experience about the actual benefit of implementing this kind of system in a power plant, both in terms of contribution to fuel savings and in terms of support to operation (more information on plant conditions, easier to access and interpret, timely available and retrievable). This confirmation is important for the system developer, in order to determine the future evolution of the product, and for the user, in order to decide about the convenience of introducing these informatic systems into his plants.

- It is confirmed that involving the user, with his plant experience and his particular needs, since the beginning of the customization process, is essential for the effectiveness of the application.

- Standardization of base HW/SW tools and integration of applications in plant automation systems are also essential points for providing expert systems with industrial characteristics.

- Capability to tailor man-machine interface to the user's requirements is a condition for the successful field use of an informatic system.

6. REFERENCES

1. A. D'Ambrosio, M. Oriati and A. Serventi. "AI in power plants: PERF-EXS, a performance diagnostics expert system", _The fourth IEEE Conference on Artificial Intelligence applications_, sponsored by The Computer Society of IEEE, San Diego, California, March 14-18, 1988

SMOP: Smart Operator's Aid for Power Plant Opitimization

R. P. PAPILLA
Southern California Edison
Rosemead, California, USA

E. J. SUGAY
Impell Corporation
Walut Creek, California, USA

ABSTRACT

SMOP is an expert system for diagnosing the cause(s) of heat rate degradation in oil- and gas-fired power plants. The system is designed to be embedded in a plant computer and to run in real-time. SMOP receives data from field instruments the control system, and Plant Performance Monitoring System (PPMS) calculations; applies the knowledge base(s) apropos to reason over the information; and displays the results of its inference in the form of recommended actions to bring the plant to a more efficient operating state. System competence was modeled primarily after the diagnostic expertise of SCE performance engineers; additional input from operators, maintenance personnel and control system vendor expert were also incorporated to enhance the knowledge base. SMOP was prototyped in a PC and is scheduled to be integrated into the PPMS MicroVAX computer at SCE's Huntington Beach Generating Station in June 1989.

1. INTRODUCTION

This project is a joint effort between Southern California Edison and Combustion Engineering (primarily Impell Corporation). Its goals relate to those of the Coal Combustion Systems Division of EPRI as described in EPRI RFP2923-3 (Development and Application of Expert Systems to Fossil Plants).

The objectives of the SMOP project can be viewed from two different perspectives. From the larger and functional perspective, the objective is to construct a problem-solving system based on a model of competence which is capable of high performance levels [1] in the domain of power plant heat rate

(efficiency) degradation. The system is generic enough to be portable, with plant specific modifications, to fossil-fired plants other than Huntington Beach Units 1 & 2.

From a software viewpoint, the objective is to develop, install and prove the utility of a knowledge-based plant optimization advisor which uses commercial expert system shells and runs in real-time at a power plant with minimum or no request(s) for data from the operator.

In conjunction with a Plant Performance Monitoring System, SMOP should provide adequate recommendations to significantly improve existing plant heat rate. It is also envisioned that SMOP will be used as a tutorial for training power plant operators.

The SMOP system is scheduled for integration into Huntington Beach Generating Station's Plant Performance Monitoring System in June 1989.

2. SMOP SYSTEM DESCRIPTION

2.1 Overview

The Smart Operator's Aid For Power Plant Optimization (SMOP) is a knowledge-based system whose goal is to help fossil plant operations and performance personnel in optimizing heat rate (BTU/KWH) by reliably, adequately and consistently diagnosing certain operator controllable losses, and some system and equipment malfunctions.

Key features of the SMOP project are:

- o Real-time expert system built with commercial expert system shell
- o Embedded and fully integrated into a plant computer system
- o Interact with operators with minimum or no request for input.
- o Utilize data from other (conventional) systems and reason over such data.

Expert system technology has been used to implement SMOP for the following reasons:

- The domain of heat rate degradation diagnosis in its entirety is computationally intractable and too complex to be amenable to structured methodologies of analysis [2]. Expert system techniques allow the representation of the cognitive processes of a plant performance expert in the diagnosis of heat rate degradation more closely than conventional techniques.

- Higher-level reasoning processes (mimicking human expertise) can readily operate on raw data as well as data already generated by extensive PPMS performance calculations, heat balance models, and the control system.

- The design, prototype coding and testing of the system is facilitated.

- Revision, maintenance and evolution of the system by non-programmers after installation is possible.

- The installation proves the utility of applying expert system technology to power plant control and optimization problems.

2.2 Statement of Function

The SMOP system is designed as an operator's aid which provides the following functions:

- Facilitates the reliable and immediate diagnosis of certain Operator Controllable Losses (OCL) and some system and equipment malfunctions in Units 1 & 2 of the Huntington Beach Generating Station (HBGS) of SCE;

- Provides justification for these diagnoses by either (i) highlighting the most probable cause in a logic fault tree graphic display or (ii) explaining the conclusion(s) obtained in the form of operator messages which enumerate data and logic used to arrive at the conclusion(s).

- Recommends corrective action prioritized by their ease of execution and relative economic impact ($/shift) to heat rate.

- o Sensitive to current state of the plant and allows the operator to pose his question by only touching appropriate areas of the screen.

2.3 Operator Controllable Losses (OCL)

The following is a list of the Operator controllable Losses which SMOP addresses.

(1) condenser system losses
(2) reheater spray flow
(3) reheat temperature
(4) main steam temperature
(5) main steam pressure
(6) stack gas exit temperature
(7) auxiliary MW losses
(8) auxiliary steam losses
(9) boiler excess air
(10) boiler carbon monoxide
(11) turbine cycle losses

In general, each of these OCL items has an associated knowledge-base associated with it. (Refer to Section 3.1, Software Architecture, for a detailed discussion of knowledge-base implementation into knowledge islands.) The losses incurred due to deviations in reheat and main steam temperature are further broken down into those associated with the temperatures being high or low.

2.4 Sources Of Knowledge

The model of competence which is coded into the SMOP system has been acquired from the following sources:

(a) Interview sessions with the performance engineer at HBGS (primary);

(b) Interview sessions with an experienced plant operator and an instrument maintenance supervisor at HBGS;

(c) Interview sessions with the combustion control system vendor's expert.

(d) HBGS Station Order HB 0-113 (revised April 9, 1986);

(e) EPRI CS-4554, "Heat Rate Improvement Guidelines for Existing Fossil Plants;"

(f) EPRI NP-4990P, "Thermal Performance Diagnostic Manual for Nuclear Power Plants."

2.5 Development and Delivery Platforms

SMOP has been developed and prototyped in a Compaq 386 machine with 4Mb of RAM, using an expert system development environment called Nexpert Object from Neuron Data Incorporated.

The SMOP system will be delivered as an embedded real-time diagnostic and monitoring system which runs as an independent process in a DEC MicroVAX II computer. The block diagram is shown in Figure 1. In the target computer, SMOP is a VAX shareable image which executes under the VMS operating system. Man-machine interface is provided through the PPMS's Taylor MOD300 CRT console subsystem which is equipped with touch sensitive screen.

3. APPROACH

3.1 Software Architecture

In terms of its architecture, SMOP is a hybrid system: it uses knowledge-based techniques to encode the diagnostic expertise for controlling heat rate, and conventional software techniques for embedding the knowledge-base into the hardware platform.

The SMOP knowledge base has been developed using a logical fault tree representation. In other words, a particular loss or a set of losses is hypothesized, then evidence is gathered to support or negate the hypotheses, and finally, the conclusion is drawn based on the nature of proven hypotheses. This methodology can be shown in the following diagram:

```
HYPOTHESIZE                          FIND EVIDENCE          CONCLUSION
  LOSS ------------------------->     TO PROVE ---------->   FOLLOWS FROM
                                      HYPOTHESIS              HYPOTHESIS
```

The knowledge base implementation is itself a hybrid system. The underlying representation of the world is modelled using objects. The orthogonal reasoning

dimension is modelled using the formalism of rules.

Declarative knowledge is used extensively to describe a generic power plant with its tangible components and equipment, and its intangible attributes, such as loss, value, etc. In this manner, the knowledge is explicit, flexible and can be easily accessed by introspective programs to reason over the knowledge content.

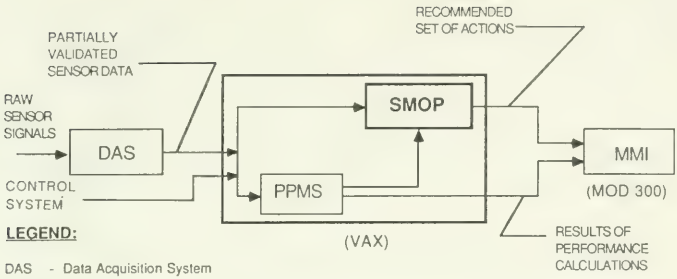
Non-monotonic reasoning is used in instances where rule revision is appropriate because the problem had changed at the next instant of time that the SMOP is run. This is achieved by remembering the state of the plant and the results of the previous inference cycle. Let us say that during one inference cycle, conditions are such that the cause of high reheat temperature is hypothesized to be a wide open (stuck) recirculation fan damper at high loads. During the next inference cycle, the process of reasoning will be biased toward the same hypothesis and conclusion. However, if main steam pressure is trending downward, within a certain tolerable bandwidth but below its expected sliding pressure value, then this new condition will trigger a revision of rules during this cycle. The ultimate result is a new hypothesis-- main steam pressure trending down-- which will be on the system's agenda.

The SMOP system architecture is modular. There are four high level architectural components to SMOP (Figure 2 shows a structure chart at this level of abstraction):

- Finite State Machine
- Signal Processing
- Input/Output Handlers
- Knowledge Bases

Finite State Machine [3]. Finite State Machines (FSM) are used to express the algorithms that determine which functions need to be activated or enabled by the system Function (or Task) Activator.

An FSM is a model of a computing device which has the following characteristics:



LEGEND:

DAS - Data Acquisition System

PPMS - Plant Performance Monitoring System

SMOP - Smart Operator's Aid For Efficient Plant Operation

MMI - Man Machine Interface

FIGURE 1. System Block Diagram

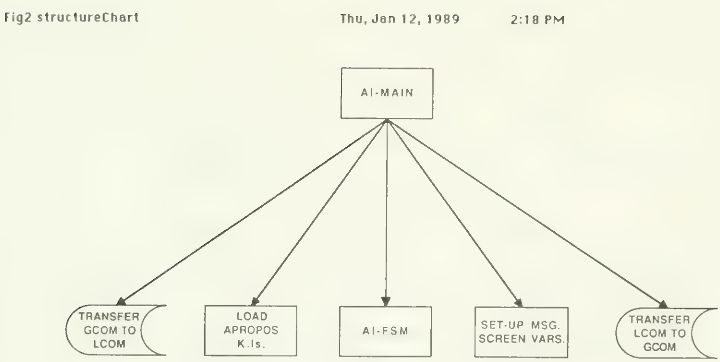


FIGURE 2. Main Structure Chart

- o it has a fixed and finite amount of memory (also called states)
- o it responds to a fixed and finite number of conditions

The FSM model which is used in SMOP is the Mealy Model. In this model, actions are associated with a tuple called a (current state):(current condition) pair. This means that an action results from a specific condition being satisfied while in a specific state. In addition, the next state of the FSM is also uniquely defined and determined by the (current state):(current condition) pair. In this case, the "states" are plant operating modes which are explicitly defined to put into context the relevance of certain heat losses.

Conceptually, the Mealy Model of a Finite State Machine can be implemented as a matrix with the following characteristics (refer to Figure 3):

(a) the rows correspond to current states, Si;

(b) the columns correspond to current transition conditions, Cj; and

(c) the matrix entries consist of the resulting actions, Tij, and the next state, Sj, associated with that (current state):(current condition), pair.

The Finite State Machine (FSM) functions as an executive process in the SMOP implementation. It prunes the list of losses down to the most probable, the diagnosis of which will yield the greatest value. Hence, the FSM has the smarts to assess the current operating mode (or state) of the plant, hold the possible losses against the CONTEXT of the current operating mode, and load only the appropriate knowledge module(s) to bear upon the problem.

The use of FSM's in processes which can be described in terms of a finite number of states and transition conditions for each of those states provides a framework which is independent of software application modules and within which each application module can be embedded to obtain function, data and control support. Further, it is a conceptual model which represents the cognitive process involved when a human expert thinks and abstracts about the diagnosis of heat rate problems in the real world. Finally, the computational implementation of a system using an FSM increases speed of execution. Essentially,

STATE \ CONDITION	C1	C2	C3	C4	C5	C6	C7	. . .
S1	S1	T12/S2	T13/S3	T14/S4	T15/S5	T16/S6		
S2	T21/S1	S2	T23/S3	T24/S4	T25/S5			
S3	T31/S1	T32/S2	S3	T34/S4				
S4	T41/S1	T42/S2	T43/S3	S4				
S5	T51/S1	T52/S2			S5			
S6	T61/S1					S6		
. . .								

LEGEND:

Si = Current State

Cj = Transition Conditions

Tij = List of Transition Actions from State i to State j

Sj = Next State

FIGURE 3. MEALY MODEL OF FSM

"meta-knowledge" is used at the top level to prune irrelevant branches from the logic fault tree.

Signal Processing. A unique feature of a real-time expert system application is the extraction of certain characteristics from the stream of real-time data. These features include qualities which characterize certain data streams using adjectives which describe degree or level (high, very high, normal, etc.), trends (trending up, trending down, constant, etc.), rate of change (increasing at a fast rate, increasing at a decreasing rate, etc.), stability (oscillating, stable, etc.), presence of spikes, etc. In SMOP, these characteristics are extracted using two methods. Trends are abstracted by subjecting the data series to a tuning controller consisting of a rectifier and a smoother [4, 5]. Statistical methods (average and standard deviation) are used to produce a qualitative estimation of the other signal characteristics.

Input/Output Handlers. The SMOP system maintains its own common area, isolated and independent from the PPMS global common. During its time slice of execution, the SMOP system maps in all the required real-time data from PPMS into its own local common. After its interference cycle, the SMOP system maps out the results from its local common back to the PPMS. Data from the control system is acquired into the PPMS CPU and treated as part of the PPMS global common.

Knowledge Bases. The SMOP knowledge bases have been developed modularly: each Operator Controllable Loss item enumerated in Section 2.3 has a knowledge base island associated with it. Hence, the knowledge required to bear upon a particular problem is self-contained within a module or "knowledge island." Further, the object world definition itself is a separate knowledge module. This means that each of the knowledge islands associated with the loss components contains only the necessary production rules and is therefore very compact. The elegance of this design lies in the fact that the knowledge islands share only one common definition of the world. This facilitates system maintenance and evolution, and avoids conflicts in knowledge declarations unless intentionally specified by the expert. This method is particularly suited to power plants where similar components occur in many instances and functional interrelationships are usually unique.

Control strategy is embedded within instances of objects. Relationships among different knowledge islands are established by dynamically revising the systems's agenda of hypotheses [6]. Declarative knowledge is used extensively to arrive at a more generic model of the world which can be used as a framework in porting the system to similar fossil units.

Objects, properties, classes, rules, and hypotheses are given names of sufficient length in order that they can be easily recognized and modified by a plant engineer porting SMOP to other units.

3.2 Man-Machine Interface

The PPMS's man-machine interface will consist of color graphics portraying portions of the plant where certain heat loss component(s) and associated data are presented to the operator. The SMOP system graphics are displayed on the very same Taylor PPMS MOD300 CRT's which are equipped with touch screens. The operator looking at the PPMS display may request either a generalized or loss-specific "help" by touching appropriate areas of the graphics.

The SMOP displays will be pre-defined, i.e., the screens are displayed by providing the interface with a unique label for each graphic file. As part of its output to the PPMS, the SMOP system maps the picture's label which results from its inference, and the appropriate remedial action message associated with the graphic. Recommended actions are displayed in the graphics which contain information relevant to the loss in question.

3.3 Integrated Operation

The SMOP system will be a memory resident task in the MicroVAX. It will have a scheduled cycle activation time of sixty seconds and a three second time slice. Without any intervention from the operator, the SMOP system automatically refreshes the available inference results regularly during its time slice of execution, and maps these results into the VAX system global common. SMOP help messages are displayed in the same graphics used by the PPMS.

The SMOP system will also run asynchronously from its scheduled period of execution upon operator request. In essence, the operator proposes a set of hypotheses by selecting one or more loss components from one of the PPMS displays on the MOD300 CRT screen. This invokes the PPMS "HELP" option, which

triggers the asynchronous execution of the SMOP system, with the agenda as specified by the operator.

4. CONCLUSIONS

The prototype of SMOP has been demonstrated to Impell and SCE management. Support for its integration into the real-time operating environment of the HBGS plant as originally planned is high. Experiments and benchmark performance comparisons are being conducted on the prototype while the system is being installed in its delivery environment.

The following enhancements to the SMOP knowledge base are planned:

- Addition of turbine and other ancillary equipment losses;

- Evolution of SMOP into a system based on causal process models [6] of the heat rate degradation problem. These are "deeper" models of domain knowledge which would enable SMOP to account for original symptoms using fundamental thermodynamic heat balances. A concomitant result of this is the ability to automatically revise and update the knowledge base when certain parameters change.

- Capability to perform smart sensor data validation during diagnosis [7, 8]. One underlying assumption in most diagnostic systems is the veracity of the data input into the expert system. In real-world applications, data is often unreliable. A truly smart system should be capable of either working with conflicting data to arrive at an adequate comprise, or at least realizing the bad quality of the data input and avoiding the pitfall of inferencing on bad data.

- Expansion to include reasoning on historical (disk-recorded) data to quantify slow equipment degradations, control stability, and operational enhancements.

ACKNOWLEDGEMENT

The authors would like to express their sincerest gratitude to the group of domain experts for this project: Juan Armas (primary source), Terry Kuntz and Ray Thompson of the Huntington Beach Generating Station, and Theo Heatherington

of Bailey Controls. They are to be commended for their untiring patience during the many hours of interviews and evaluation of SMOP's performance. Credits are due to Uday Shendrikar, Carman Winarski and John McKinley of the Engineering, Planning and Research Department of SCE, and Pete Geister of the Huntington Beach Generating Station. They are instrumental in initiating and supporting the continued work on this project. This development project is being funded jointly by SCE and Impell Corporation's Technology Development Program. Gratitude is also due to Jim Rivello, Larry Metcalf, and Scott Damesek of Combustion Engineering for their support in software implementation on the PPMS MicroVAX.

REFERENCES

1. Keravnou, E. T. and Johnson, L. Competent Expert Systems, A Case Study in Fault Diagnosis. New York: McGraw-Hill Book Company, 1986.

2. Partridge, D. Artificial Intelligence: Applications in the Future of Software Engineering. London: Ellis Horwood Limited, 1986.

3. Krishnamurthy, E. V. Introductory Theory of Computer Science. New York: Springer-Verlag New York, Inc., 1985.

4. Astrom, K. J. American Control Conference, Boston, MA, 1985, pp. 1515-1519.

5. Automatica. Special Issue on Adaptive Control, Vol. 20, 1984, pp. 449-716.

6. Hayes-Roth, F., Waterman, D. A. and Lenat, D. B. (eds). Building Expert Systems. New York: Addison-Wesley Publishing Company, Inc., 1983.

7. Chandrasekaran, B. and Punch, W. F. "Data Validation During Diagnosis, A Step Beyond Traditional Sensor Validation." Proceeding of the AAAI, Seattle, Washington, July 1987, pp. 778-782.

8. Hashemi, S., et al. "Expert Systems Application to Plant Diagnosis and Sensor Data Validation." Proceedings of the Sixth Power Plant Dynamics, Control and Testing Symposium, Knoxville, Tennessee, 1986, pp. 65.01-65.13.

Coal Quality Advisor for Coal Buyer

B. R. ARORA
Houston Lighting & Power Co.
P.O. Box 1700, Houston, Texas 77251, USA

J. P. RACINE, R. H. SIROIS, and G. A. FINN
Stone & Webster Engineering Corp.
245 Summer Street, Boston, Massachusetts 02210, USA

R. S. HANNA, E. E. KERN, and A. L. BUFFINTON
Houston Lighting & Power Co.
P.O. Box 1700, Houston, Texas 77251, USA

ABSTRACT

Development of a personal computer based expert system for coal purchasing application has recently been completed by a joint Houston Lighting & Power and Stone & Webster Engineering Corporation project team.

This paper describes various tasks leading to the development of the system and results of testing and validation exercises carried out to check the system intelligence. The intended application of the system is to assist a utility coal buyer to make detailed assessment of cost impacts of burning a given coal in a given plant. Detailed coal analysis obtained from potential vendors is required as input for the system assessment which takes into account the fuel characteristics and its behavior during handling, combustion and ash disposal as they impact electricity production costs.

The paper also discusses some lessons learned in managing a project of this nature.

INTRODUCTION

An expert system, Coal Quality Advisor (CQA), for use by a utility coal buyer has recently been completed by a joint Houston Lighting & Power Co. (HL&P) and Stone & Webster Engineering Corp. (S&W) team.

The system runs on a personal computer and its intended application is to assist a coal buyer in making detailed assessment of cost and performance impacts of using a candidate coal in his plant. A pulverized coal fired plant with a baghouse and a wet limestone scrubber was modeled for this application. The capability exists to replace the baghouse with an electrostatic precipitator in order to model different pulverized coal fired plants.

The personal computer based EXSYS[1] software was used in the system development. The heuristic knowledgebase assembled in the form of rules is comprised of expert experience of the participating organizations.

Testing and validation of the system included verification with cost and performance data from HL&P's W. A. Parish Unit #8.

TECHNICAL PROBLEM DESCRIPTION

A utility coal buyer must consider all costs of using a candidate coal. This includes its purchase price and transportation plus all plant impact costs due to its use. This implies that detailed cost impacts of coal quality on plant performance must be assessed. Beyond cost implications, the buyer should know that the characteristics of the candidate coals are compatible with the design of the plant.

PROJECT ORGANIZATION

A joint HL&P/S&W team was formed with the following individual roles -

- Domain Experts from both organizations
- Knowledge Engineer
- System Integrator
- Overall Engineering Coordinator
- Project Manager

MAJOR TASKS

The project was executed in three major tasks as follows -

a. Functional Specifications Development
b. System Development & Integration
c. Testing & Validation

In the functional specifications development task, the various subtasks included hardware selection, shell selection, detailed layout of input and output screens, on line queries specification, models functionality, reporting formats and a rapid prototype development.

In the system development task, an overall engineering logic flow chart was first developed. Algorithmic models were developed and expert interviews were conducted to establish heuristic rules. Programming and logic coordination then proceeded to integrate system functionality. Expanded prototypes were used to monitor system development.

In the testing and validation task of the developed system, we performed numerous test exercises to check the system response. Several cycles of updates of system programming were required to bring system performance up to a level where experts and end users have confidence in its results. The system is currently in use in the Company.

---

[1]EXSYS is a production rule based expert system development package by EXSYS Inc., Albuquerque, NM.

EXPERT SYSTEM CONFIGURATION

Predicting the effects of coal constituents on the performance of critical plant systems, such as the boiler and pulverizers, requires the assimilated knowledge of experts in several areas. Specifically, these areas include coal chemical analysis, plant engineering and design, plant operations, and operation and maintenance cost analysis. Because the fundamental goal of the Coal Quality Advisor is to assist the coal purchasing engineer in the thorough evaluation of

candidate coals/coal blends, each of these critical knowledge components must be integrated into the expert system.

Knowledgebase components of the Coal Quality Advisor are:

Coal Analysis
Plant Operations
Plant Engineering, and
Cost Analysis

These must be accessible from a common source.

The functional requirements of the expert system follow directly from the above definition of knowledgebase components. First, a verification of the coal analysis received from the potential coal vendor must be made. This requires the expertise of one who is familiar with the wide range of coals which may be offered. This involves not only a check of the arithmetic consistency of the analysis, but also a comparison of the coal constituents with coals of similar classification and ash characteristics.

With the verification of the coal analysis completed, an evaluation of the coal's impact on the plant may be made. Typically, this is a two-phased evaluation. First, a qualitative assessment is made, highlighting the areas of the plant which are most likely to be affected by the coal properties. The expert's judgment in identifying these areas is critical to the efficient and accurate evaluation of the coal. Oversights in this phase can result in unforeseen impacts on operation and maintenance costs. Second, each critical area of the plant is investigated in detail. At this point engineering and environmental limitations are checked. This often requires specific plant operation knowledge, relating operating troubles with various coals or coal blends. Only after these steps are performed can a decision to purchase a coal, or provide a specific coal blend, be confidently made. This general solution approach is shown in Figure 1.

The requirements placed on the organization of this intelligence within the knowledgebase results in a system that is modular both in structure and in function. Thus, the Coal Quality Advisor is composed of a number of separate programs, each interacting with its counterpart programs automatically. The individual programs are used to generate plant-specific operating data which can be evaluated by the expert system rules. The commercial software which is used to control not only the inference across the rules, but the execution of each of these individual models, is EXSYS. The individual system modules are listed below, and are described in subsequent paragraphs.

- User Interface
- Coal Analysis Verification Rule Base
- Plant Impact Rule Base
- Plant Models:
  - Coal Handling System
  - Pulverizer System

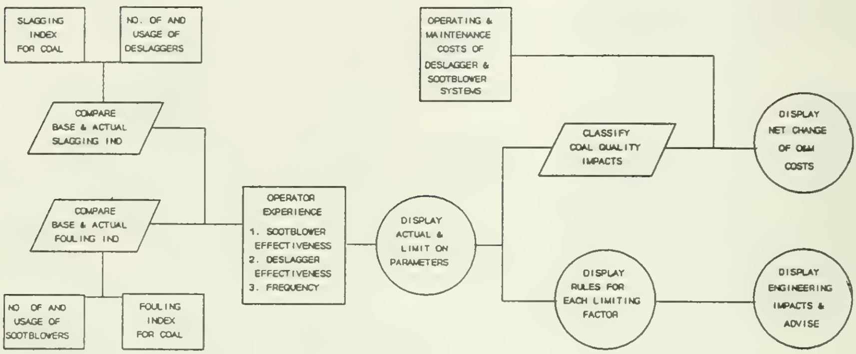


Figure 1. Representative Solution Process

Boiler: Furnace and Convection Pass
Draft System
Sootblower System
Ash Handling System
Particulate Collection System
Scrubber System

- Cost Evaluation Facility
- Report Generation
- Miscellaneous File Handling

The primary User Interface is a menu-driven data entry program which provides access to all system functions. Data entry is accomplished through the use of convenient input templates for each group of data required: coal analysis, plant operating parameters, and reference cost data. On-line menu-specific and datafield-specific help is available where appropriate. The user has access to a coal blending optimization module through this interface.

The task of verifying the accuracy of the candidate coal analysis is accomplished through the Coal Analysis Verification Rule Base. This is a distinct rule base which takes as input the coal analysis values provided by the user through the data entry forms. The system establishes the coal rank (based on ASTM D388 guidelines), ash type, and determines ash fouling and slagging characteristics. It then proceeds to compare the provided analysis values against those expected for similar types of coal and coal ash. Any discrepancies are displayed to the user at the end of the verification session.

The Plant Impact Rule Base contains rules to evaluate the effect of the candidate coal/coal blend on the plant systems. The rules are organized into three general categories: data verification, data interpretation, and impact assessment. Data verification occurs prior to calling any of the plant system models. A check is included to ensure that the values being provided are reasonable from an engineering perspective, and comply with the expected ranges for the coal type and specific plant which has been modelled. Data interpretation occurs after the plant models have been evaluated. The interpreted data (in an "English sentence" format) is then used to assess the impact of the coal on the plant systems. The impact assessment for each of the plant systems is classified by severity level and displayed to the user with appropriate recommendations.

The Cost Evaluation Facility contains procedures which determine the relative O&M cost impact of burning the candidate coal/coal blend. Each parameter used in the evaluation is maintained by this facility (actually a series of rules) throughout the plant impact session. At the end of the session, these parameters are combined mathematically to provide the overall relative cost impact.

Throughout the rule evaluation procedure, the user is provided with several opportunities to query the knowledgebase in order to determine the logic behind any questions being asked, or to determine the source of any of the numeric values and qualitative interpretations provided by the expert system. At the end of the evaluation, the user may obtain explanations, by using a similar trace-back facility.

SOLUTION APPROACH TO TECHNICAL PROBLEM

The objective of evaluating coal quality on the technical aspects and Operation & Maintenance (O&M) cost impact for a specific coal fired unit within the HL&P system was a challenging assignment in many respects.

The development of the Coal Quality Advisor included the following features:

a. Blending of up to 5 candidate coals to a specified mix or to achieve a specified quality for the blend (i.e., sulfur, ash, heating value).
b. Classifying the coal (blend) to permit assessment in various components.
c. Determination of slagging and fouling indexes, based on coal rank.
d. Evaluating the required performance against the given limits for:
   - Coal handling system
   - Pulverizer System
   - Boiler: Furnace & Convection Pass
   - Draft System
   - Sootblower System
   - Ash Handling
   - Particulate Collection System
   - Scrubber System
e. Determination of O&M costs and the net heat rate change for a candidate coal relative to a given base coal.
f. Configure initially for Unit #8 at HL&P Parish Plant, but enable evaluation of other coal fired units in the HL&P system with minor changes to the Coal Quality Advisor system.

The entire physical system from coal handling through gas cleanup was evaluated, although the focus of the program was the boiler.

Prior to commencing the program development, meetings were held between technical and management personnel involved with the project to define the exact scope, the level of detail and the information required to meet the needs of the HL&P coal purchaser.

The foundation of the system was initially described on logic diagrams which were prepared for each major plant system referred to in the above item (d). These diagrams itemized the preliminary requirements for:

- Input items to support the analysis;
- Calculations of parameters judged important;
- Operation and Maintenance personnel comments to augment the technical analysis and rule base;
- O&M cost components;
- Displays of input and output data and rules;

A representative diagram used for this purpose is depicted in Figure 1.

On large, comprehensive programs, such as the Coal Quality Advisor, it is important to establish the above to ensure that the development team knows:

- What information is available or needs to be retrieved;
- The level of detail to which calculations will be performed, and their technical basis;
- The type of interaction that will be required during the execution of the project;
- A means to chart progress as the system is developed.

The approach of first developing a system to be applied for a specific generating unit and then extending it later to a more generic expert system has the following benefits:

- The confidence level of results is more likely to be higher since site specific information is known;

- The cost of development is lower, since only a specific data base is required;
- The impact comparison of one coal to a base coal in a specific unit is more easily achieved than developing an assessment based on literature information.

Once such a development plan was jointly approved, the input files, calculation routines and rule base were developed.

The following represents an overview of the level of detail of the data used and results attained.

<u>Coal Verification</u>

Data:

- Proximate analysis
- Ultimate analysis
- Sulphur forms
- Minerals of ash
- Acid soluble alkali
- Ash fusion temperatures
- Trace elements
- Miscellaneous
    - Hardgrove grindability
    - Equilibrium moisture
    - Quartz content

Results:

- Check on sum of and acceptable range of constituents
- Classification of coal using ASTM Standard Classifications
- Slagging Index, Fouling Index and classification rules on:
    - HGI (Hardgrove Grindability Index)
    - T250 temperature (Ash Viscosity)
    - Ash fusion temperature differentials

<u>Boiler</u>

Since the boiler is considered the critical component when evaluating potential coal (blends), the Coal Quality Advisor performs a thorough analysis and addresses numerous factors affecting operation and maintenance. The foundation of this is a computer routine developed by Stone & Webster. The routine evaluates up to 25 heat transfer sections comprised of five kinds of heat exchanges.

The program is suited for estimating the performance of a unit which has conditions changed from a model (known or measured) case. The scope of evaluation includes changes in: fuel, operating pressure/temperature, surface area or configuration, and steam flow.

The specific functions performed are:

For a fossil fuel having a given ultimate analysis, combustion calculations are performed which define combustion air and flue gas quantities.

Boiler efficiency is calculated by the Heat Loss Method, with given radiation, unburned carbon, and manufacturers margin losses, and any credits which may exist.

Iterative heat transfer calculations are made which account for direct furnace radiation, non-luminous radiation, convection, and absorption by enclosure walls for a multitude of conditions normally found in boilers. The capability to evaluate bisector or trisector air heater performance also is an integral part of the program.

Physical conditions of varying furnace exit gas temperature, radiant section duty, spray water flow quantities, air heater duty and outlet gas temperature, boiler efficiency and heat input to the furnace, boiler duty and steam/gas flow heat balance are simultaneously performed to mathematically simulate the performance of the boiler. Gas side draft loss is also calculated for the boiler components.

Coal Storage Systems

The coal storage systems considered are comprised of live and boiler silo storage. The pile capacity is considered ample; therefore, it was not evaluated.

Based on given transport capacity to plant the program determines the number of days storage with the given coal and compares against a minimum time desired. The respective fill intervals are also evaluated. The analysis assumes that the three other coal fired units at Parish Station each use the same amount of coal as is being consumed by Unit No. 8.

Coal Transport Systems

This section is separated into coal transport, unloading and stack-out, dumper/feeder, conveyors and reclaim systems into the plant for four units.

As with the above systems, the reclaim system operating capacity is calculated and compared against system capacity.

Pulverizer System

The scope of this system is comprised of the pulverizer feeder, pulverizer and burner transport lines. Based on the specified number of pulverizers (mills) in service, the Program tests the required coal flow per mill against the maximum feeder capacity.

Since the wear of pulverizer parts and burner lines can be serious reliability factors, a detailed appraisal of these components is made. The relative wear between a base coal and the candidate coal is affected by coal and air flow, abrasion index and number of mills in service.

Burner Liberation

Exceeding the allowed firing rate in a burner can result in reliability and maintenance problems. Since the burners on Unit 8 at Parish Station are rated for a maximum liberation, it is important to evaluate the planned operation against limits.

Boiler Furnace

Varying coal constituents/properties have a major effect on boiler performance. Many of these factors deal with the combustion process. Through experience, a

number of furnace geometric factors and operating limits have been developed to impose design limits as a function of coal properties: slagging and fouling indexes, for example.

Boiler Convection Pass

The major factors related to the convection pass of the boiler as it pertains to coal quality are fouling potential, flue gas velocity, and tube metal temperatures in respective sections.

For fouling coals, retractable sootblower usage is adjusted proportionately. Also, permissible clear lateral tube spacing criteria are established as a function of flue gas temperature.

Flue gas velocity is an extremely important factor in establishing erosion potential for a given coal. Combined with abrasion index and ash weight relative erosion rates are established. Heat transfer analysis model generates respective steam temperatures and spray quantities in the boiler to evaluate against design limits.

Air Heater Average Cold End Temperature

The sulphur content in the candidate coal serves to establish the criteria for the air heater average cold end temperature.

This factor indirectly provides the Coal Quality Advisor user with information on whether more auxiliary steam for steam coil usage will be required.

Ash System

Of the ash produced from the combustion of coal, a portion becomes bottom ash, a portion is captured in the economizer hopper and the balance is designated as flyash, entering the baghouse or electrostatic precipitator. Although the total ash flow must be the sum of the above, the user has the flexibility to proportion the ash going to each zone to deal with uncertainty.

Since the ash systems have specific tonnage limitations, the Coal Quality Advisor compares the actual ash flow to the specified limits.

Fans

The FD and ID fan system typically are designed to provide for adequate combustion air and proper evacuation of flue gas from the plant.

Although the probability that an operating limit will be reached with the use of any candidate coal is rather low, the check is still made. The important implication for evaluating the fans is the relative horsepower requirement between coal or coal blends. For that reason, fan performance curves are used to determine relative pressure differentials as a function of flow, and drive horsepower differentials are established and compared to base conditions. Per HL&P specification, the horsepower differentials are equated to quantities of incremental coal flow, rather than differential energy consumed, based on net plant heat rate and capacity factors.

Baghouse

The coal Quality Advisor evaluates the actual operating conditions against the environmental and downstream component contraints to check for suitability.

Electrostatic Precipitator

The alternative to a baghouse in the HL&P system for particulate collection is a cold side electrostatic precipitator. Collection efficiency as a function of coal/coal ash properties is not easily established since the variation of ash resistivity as a function of many constituents can be very large. Consequently, collection efficiency can be quite variable.

The Coal Quality Advisor approaches the problem by determining ash resistivity only as a function of coal sulphur content. The efficiency is then determined through the Duetch-Anderson Equation.

Similar to the baghouse, the emission rate is determined and tested against specified limits.

Scrubber System

Designed to capture sulfur dioxide, this system is evaluated by CQA to determine the sorbent rates at full boiler load, SO2 content and varying system absorption efficiency requirements.

Since HL&P has experience with Dibasic Acid (DBA) addition, a usage calculation of their derivation is utilized in CQA. In applications with DBA, the scrubber utilization factor is increased. Other factors assumed to get DBA usage are a L/G ratio a PH and a pump factor.

Other factors considered as limits are the range of S content, maximum Cl content and Cl/S ratio.

O&M Costs

In developing the methodology to determine the Operation and Maintenance cost changes with coal quality on this unit, it became apparent that a suitable database did not exist at HL&P or elsewhere.. The primary reason for this is considered to be the wide variation in plant design and O&M practices of U.S. utility industry. Consequently, another approach was developed.

The basis for the O&M cost methods is a percentage of equipment capital costs for each major component. Considered for comparison is a 500 MW plant firing Western coal. The costs were proportioned to Parish No. 8 reported total costs, with an estimated value for coal yard work.

Generally, for those items whose O&M costs vary with some coal property, the change in cost is calculated based on a ratio of direct cost, overhead, and either proportional or inverse to the magnitude of variance between the base and candidate coal. The exception is the coal handling system which has a known variable O&M cost.

This method is considered valid for relative O&M costs only, not absolute values. The components considered are:

Coal Handling
Pulverizer System
Furnace
Draft Systems and Air Heaters
Ducts
Sootblower System
Boiler Heating Surfaces
Ash Handling Systems

Particulate Collection System
Flue Gas Desulfurization (FGD) System

The change in auxiliary power resulting from a change in coal is evaluated for the major components only, not for small or fractional horsepower systems. The F.D. and I.D. fans and pulverizers are evaluated. Fan performance curves are used to predict fan power requirements. The relative differences in power requirements are reported as a change in coal tonnage, based on a recalculated heat rate and a given main steam and reheat flow.

Testing and Validation

The testing and validation of a program must subject it to as many potential variables as possible. For the CQA, several coal analyses were selected with each containing a characteristic that should be identified by the program. High sulfur, low grindability, high iron in ash, etc.

The following is an edited example of a printout from the program. Editing has simply deleted items that are repeated to permit each section to stand alone if it is desirable. For example, one might want only the impact on economics but should be compelled to at least see the problem areas behind the evaluation of O&M and recognize that the program is an Advisor, not a Designer. The "user", a person using the program to evaluate a coal, must heed advice given or the program is useless.

EDITED OUTPUT OF A SAMPLE TEST EXERCISE

A printout of a program run follows with "COMMENTS" inserted:

- COAL VERIFICATION -

Supplier: SUPP3
Coal Identification: TEST3
Verification Date: WED Mar 08 14:20:47 1989

This coal has the following characteristics:

Coal classification as per ASTM D 388 is Subbituminous
Coal sub-class as per ASTM D 388 is A
Ash classification is Bituminous
Ash fouling classification is medium - Fouling factor: 0.11
Ash slagging classification is medium - Slagging factor: 0.48

PROXIMATE ANALYSIS

No problems have been detected with the Proximate Analysis.

ULTIMATE ANALYSIS

The range of values for the sulfur content from the ultimate analysis should be provided. Value(s) to check:

Minimum Sulfur 0.00
Maximum Sulfur 0.00

COMMENT: Only average values were provided in the input to the program. A range of values permits a look at maximums and minimums that might occur but not necessarily in the same analysis.

Based on the given HHV, the coal is non-compliance. Value(s) to check:

Average Sulfur	1.05

COMMENT: Rule establishes 1.20 pounds SO2 per MMBtu. Program calculated 2.16 pounds.

SULFUR FORMS

The average values for the sulfur forms should be provided. Value(s) to check:

Average Organic	0.00
Average Pyritic	0.00
Average Sulfate	0.00

The sum of the sulfur forms should be equal to the sulfur content as recorded in the ultimate analysis.

ASH MINERAL ANALYSIS

The undetermined value should not exceed 2%. Question the source of the analysis. Value(s) to check:

Undetermined (average)	2.91

COMMENT: Experience indicates and rule states that "undetermined" can be held to plus or minus 2%, which is exceeded here.

ACID/WATER SOLUBLE ALKALI

The acid soluble basis should be used to calculate the fouling index of a lignite or subbituminous coal. It has not been provided in this case. Therefore, the sodium and potassium contents from the mineral analysis will be substituted. Value(s) to check:

Acid Soluble Basis

Na20 (average)	0.00
K20 (average)	0.00

COMMENT: Acid or water soluble alkalis are not usual in commercial analyses but are most reactive. Comment alerts user to their value.

MISCELLANEOUS

The average values for the miscellaneous items should be provided. Value(s) to check:

Average Equilibrium Moisture	0.00
Average Coal Size	0.00

It is not typical that the HGI be less than 45. Value(s) to check:

Average HGI	43.00

COMMENT: Hardgrove Grindability Index is essential to establish pulverizer performance. Rule sets minimum expected as 45.

The moisture at which the HGI was determined is needed to help verify the HGI value provided in the analysis. Value(s) to check:

HGI Moisture	0.00

COMMENT: In western and lignitic coals HGI varies considerably with moisture. Index at moisture expected in grinding zone is important.

ASH FUSION TEMPERATURES - REDUCING

No problems have been detected with the Reducing Temperatures.

ASH FUSION TEMPERATURES - OXIDIZING

No problems have been detected with the oxidizing temperatures.

COMMENT: Ash fusion temperatures on a reducing and oxidizing basis are essential for furnace design since both atmospheres can exist in different areas of the furnace and under varying firing conditions. Lignitic ash slagging index is determined using these temperatures.

PROXIMATE ANALYSIS

All Values Percent WT

Parameter	Ave	Min	Max
Moisture (total)	17.73	0.00	0.00
Volatile	31.14	0.00	0.00
Fixed Carbon	43.14	0.00	0.00
Ash	7.99	0.00	0.00
HHV (Btu/lb)	9712	0	0

ULTIMATE ANALYSIS

All Values Percent WT

Parameter	Ave	Min	Max
Moisture (total)	17.73	0.00	0.00
Carbon	56.95	0.00	0.00
Hydrogen	3.56	0.00	0.00
Nitrogen	1.29	0.00	0.00
Chlorine	0.08	0.00	0.00
Sulfur	1.05	0.00	0.00
Ash	7.99	0.00	0.00
Oxygen (difference)	11.35	0.00	0.00

COMMENT: Since, in analysis procedures, proximate fixed carbon and ultimate oxygen are determined by subtracting "determined" values from 100 the rule states that totals must be 100 or the analysis should be questioned. No problem with analysis shown.

SULFUR FORMS

All Values Percent WT

Parameter	Ave	Min	Max
Sulfate	0.000	0.000	0.000
Organic	0.000	0.000	0.000
Pyritic	0.000	0.000	0.000

ASH MINERAL ANALYSIS

All Values Percent WT of Ash

Parameter	Ave	Min	Max
Si02	43.61	0.00	0.00
A1293	18.36	0.00	0.00
Ti02	0.95	0.00	0.00
Fe203	12.87	0.00	0.00
Ca0	7.42	0.00	0.00
Mg0	2.09	0.00	0.00
Na20	0.30	0.00	0.00
K20	1.00	0.00	0.00
P205	2.76	0.00	0.00
S03	7.73	0.00	0.00
Undetermined	2.91	0.00	0.00

COMMENT: Ash mineral analysis used in determining ash slagging and fouling indexes. Used in estimating abrasion and erosion indexes.

FUSION TEMPERATURES

All Values in Degrees F

Parameter	Ave	Min	Max
Reducing Atmosphere			
Initial Deformation	2102	0	0
Softening	2228	0	0
Hemispherical	2282	0	0
Fluid	2336	0	0
Oxidizing Atmosphere			
Initial Deformation	2282	0	0
Softening	2318	0	0
Hemispherical	2354	0	0
Fluid	2444	0	0

TRACE ELEMENTS

COMMENT: Trace element analysis is sometimes requested for use in environmental studies. Fluorine has been a scrubber corrosion problem in the southeast. Purchaser should request data if required.

OTHER ITEMS

Parameter	Ave	Min	Max
Hardgrove Grindability	43.0	0.0	0.0
Equilibrium Moisture	0.00	0.00	0.00
Quartz Content (%WT)	0.00	0.00	0.00
Coal Size (inches)	0.00	0.00	0.00

COMMENT: Quartz in ash can be estimated from mineral analysis. Abrasion and erosion result largely from presence of quartz and pyrite in coal as well as alumina in ash.

COAL IMPACT RESULTS

COMMENT: This section of the program produces the impacts that coal of the input analysis are expected to have on Engineering and Economic factors applicable to the specific boiler being considered.

- ENGINEERING -

Coal Name: TEST3
Supplier: SUPP3
Date: Wed Mar 08 14:25:13 1989

This coal (blend) has the following characteristics:

Coal classification as per ASTM D 388 is subbituminous A
Ash classification is bituminous
Ash fouling classification is medium -- Fouling factor 0.11
Ash slagging classification is medium -- Slagging factor 0.48

The following impacts are crucial and are likely to result in the rejection of the coal or blend as evaluated:

Burner operating problems anticipated. Iron content 12.87 (% wt ash) is too high. This has been proven to cause "eyebrows". (Problem ranges have been reported to be as high as 15%.)

COMMENT: Based on performance of base unit the iron in the ash is limited to 7.5% (by rule). Note that this is rated as "crucial" impact and user must exercise caution. Vendor should provide information on frequency of occurrence and operating department consulted.

The following impacts are very important and may result in the rejection of the coal after thorough engineering review. Based on the following, this coal should be closely scrutinized by operations, engineering, and maintenance. Seek the advice of outside consultants if necessary. The impact is likely high for significant capital expenditure or possible derating to achieve satisfactory operation.

COMMENT: This section deals with impacts that are "very important" based on rules. Tube erosion in the convection pass is stated as a ratio of erosion index of proposed coal to that of the base coal. In this example erosion is estimated as six times that of the base coal. Program permits access to this calculation as well as others.

Tube erosion in superheater (Section 9) is 607% greater than with base. Cl/S ratio 0.076 has exceeded maximum ratio 0.0375. Cl content 0.08 has exceeded maximum 0.03. S content 1.05 has exceeded the maximum 0.8. Scrubber design/operation is likely to be affected by the use of this coal (blend).

COMMENT: Cl/S ratio in unit scrubber design establishes a maximum value. Program calculates and comments if value is exceeded. Cl and S maximums are also established in rules based upon scrubber design.

NOx emissions may be exceeded with this coal (blend). Expected NOx emissions is 0.948529 (lbs/10e6Btu). Changes in operations or equipment may be effective in bringing NOx down to compliance levels.

COMMENT: Based upon experience, a value for NOx is established for the "base" coal. Since the fuel nitrogen is the major contributor to NOx emission the ratio of its value in the candidate coal to that of the "base" coal is used to approximate a level of NOx emission for the candidate coal compared to the "base" coal.

CaO content 7.42% is outside of acceptable range (19% to 30%). This is a difficult and costly problem to correct, and will have detrimental effects on the fly ash sales and landfilling.

COMMENT: In this version of the program a range of CaO in the ash was established for Western, lignitic ash. User should consult with those responsible for ash sales to be used in concrete or roadway base.

Waterwall sootblower use required 1 (blows/shift). This is different than that of the design coal ash by -1 (blows/shift). This may also result in reduced steam temperatures. If the ability to decrease sootblower use is not anticipated, the Furnace Exit Gas Temperature reference temperature (CFEGT) should be modified by adjusting: -50 (F)

COMMENT: In this case the "base" or design coal is rated high slagging and severe fouling and the candidate coal is medium in both indexes. The lower slagging index indicates operation of wall blowers one less time per shift. The reduced slagging indicated and resultant higher absorption in the furnace may result in a lower FEGT and lower steam temperatures. If wall blower use is not to be reduced with the candidate coal then the "base" coal FEGT should be reduced where it is used in additional evaluations of the candidate coal.

Potential for excessive slagging. FEGT 2176.51(F) has exceeded the 2078(F) limit. This could tax the superheater/reheater metal temperatures, spray capacities, ash handling system, and furnace wall sootblower systems. Increasing excess air has sometimes been effective in reducing furnace slagging. Adjust FEGT reference (CFEGT) by -4 degrees (f) per % added to excess air.

COMMENT: The limit on FEGT in this case is established as 2078F which is 150F below ash softening temperature H=W on a reducing basis. Boiler model has calculated FEGT as 2177F and warning appears. Possible effect on unit operation is flagged for user. Increase in excess air to control slagging can be estimated by the correction (CFEGT) at a reduction of the FEGT by 4 degrees F per % excess air increase. Ten percent increase in excess air lowers FEGT approximately 40 degrees F to 2137 F.

Tube spacing in superheater too low. At a gas temperature of 1737.5(F), given tube spacing 2.88 (inches) is less than the reference tube spacing by 69.33%. Fouling

is likely to occur, resulting in very rapid localized erosion, loss of heat absorption, and jeopardizing metal temperatures.

COMMENT: Suggested tube spacing for fouling or non-fouling coal at various gas temperatures throughout a Combustion Engineering unit is published in EPRI Report CS-4283. This suggested spacing is incorporated in this item and for the balance of the sections through the economizer. The statement here says that 2.88 inches clear is 69.33% less than suggested or 30.67% of suggested. Suggested then becomes 9.39 inches with a medium fouling coal ash. Although recommended by the manufacturer, we consider the suggested clearances to be excessively conservative but should guide user to discussion with Engineering or Production personnel.

The following impacts are assessed as somewhat important. Appropriate departments should be consulted in order to establish the need for further evaluation. Such factors as history, frequency and severity of problems, projected unit loading, duration and size of contract deserve consideration.

COMMENT: The boiler model has calculated gas weights and volumes. The existing clear tube spacing has been an input for this unit and the program now calculates gas velocities throughout the unit and "flags" any that exceed base unit design on base coal.

The following impacts are considered less important; however, the decision to proceed should be predicated on available knowledge and experience with the specific component is affected. There is a marginal chance that this criteria will affect the decision.

Coal piping wear penetration increased by a ratio of actual/design of 3.73. Increased wear at elbows is likely. Spare mill is not available.

COMMENT: Program has calculated the "base" coal abrasion index and that for this coal. The ratio of indexes is 3.73 indicating that this coal will wear the coal-air piping almost four times more rapidly than the "base" coal.

The following engineering parameters have been used throughout this study.

COMMENT: The following tabulation provides performance data for the user to consider in comparing coals. Estimated heat rate, boiler efficiency, coal consumption, auxiliary power, etc, can be evaluated in considering coal quality.

Maximum mill capacity (tons/hr)	69
Required coal flow (pph)	570000
Plant capacity factor	0.75
Mills in service	6
Coal flow per mill (tons/hr)	48
Percent base mill	92
Total number of mills avail. (including spares)	6
Pulverizer horsepower input	509.28
Superheater (Section 9) gas velocity in (ft/sec)	69
PA temperature (F)	715
Superheater (Section 8) gas velocity in (ft/sec)	56
PA flow (lb/hr/mill)	197281
Reheater (Section 5) gas velocity in (ft/sec)	45
Reheater (Section 6) gas velocity in (ft/sec)	45

FD horsepower	1775
Economizer (Section 11) gas velocity in (ft/sec)	54
ID horsepower	3339
Screen (Section 7) gas velocity in (ft/sec)	64
Air flow (lb/hr) uncorrected	5197964
Primary air to fuel ratio (lb air/lb fuel)	2.08
Air flow (lb/hr), air heater, including leakage	5513777
Excess air (%)	23
Fuel flow (pph) from boiler calculations	570000
Annual fuel flow (tons/yr) @ 0.75 cap. factor	1872451
Gas temp. out uncorrected (F)	327
Bottom ash flow (pph)	23019
Gas temp. out corrected (F)	317
Fly ash flow (pph)	36831
Plan area heat release rate actual (Btu/hr ft2x10e6)	1.62
Lowest H=W softening temp (F), used in FEGT check	2228
Volumetric heat release (liberation) actual (Btu/hr ft3x10e3)	9.24
Furnace exit gas temperature (F)	2177
Temperature of air into air heater (F)	102
Flue gas flow (pph)	5720216
Limestone usage rate (tons/hr)	12
DBA required (pph)	0.0
Unburned carbon (lbs per 100 lbs coal)	0.01
Boiler efficiency (%)	87.91
Approximate net heat rate (Btu/kw-hr)	10265
Limestone cost ($1000)	761.5

COAL IMPACT RESULTS

\- ECONOMIC -

COMMENT: Test identification, coal characteristics and impacts developed in the Engineering section are printed here in the economic evaluation since most directly affect O&M and will appear as a warning to the user.

COMMENT: The O&M costs, not available in detail for this unit, are, in this case, the total yearly dollars distributed essentially in accordance with the detailed breakdown available from a unit burning a coal similar to the "base" coal. The amounts in the "base" coal cost breakdown are then adjusted for such items as difference in coal flow, abrasion index, power consumption, etc. The "Differential Power Costs as Equivalent Coal Flow" indicate the amount of coal burned to generate more or less power than that consumed when burning the "base" coal.

Relative to O&M costs reported in 1987 dollars, (x1000 $/yr):

Coal Handling:	-2.56
Pulverizers:	2280.42
Boiler Pressure Parts:	-50.75
Burners and Ignition Oil System:	-5.23
Auxiliary Fuel:	0.00
Soot Blowers:	-18.38
Air Preheaters, FD & PA Fans:	24.75
Draft Ducts:	196.90

Induced Draft Fans:	38.68	
Solid Waste System:	310.91	
Particulate Removal:	609.86	
Scrubber:	1403.58	
TOTAL CHANGE IN O&M COSTS	4788.19	(x1000 $/yr)
ANNUAL FUEL FLOW	1872450.79	(tons/yr)
EFFECTIVE COST	2.56	($/ton)

Differential Power Costs as Equivalent Coal Flow (tons/yr):

Pulverizers:	2110.47	
Forced Draft Fans:	-307.97	
Induced Draft Fans:	-949.03	
TOTAL DIFFERENTIAL POWER COSTS	853.47	(tons/yr)

Approximate Net Heat Rate (Btu/kw-hr)	10265
Approximate Net Heat Rate (Btu/kw-hr) base coal	10486
Boiler Efficiency (100 - Losses)	87.91
Boiler Efficiency (100 - Losses) base coal	86.61

COMMENT: The approximate net heat rate is developed from that with the "base" coal through a formula developed by a turbine-generator designer/manufacturer. Boiler efficiency is calculated for this coal by the boiler model in the program.

CONCLUDING REMARKS

This was the first expert system development project initiated by HL&P to be put to use (after a confidence building testing and evaluation period). Its success is attributed mainly to very cooperative working relationship among project participants.

There is a general lack of experience in this type of project and we feel valuable experience was gained by all participants. We summarize below some suggestions that may contribute to the success of an expert system development project:

1. A very specific objective must be defined to serve a particular user application. The user must then participate fully in the initial important task of detailing functionality of the system.

2. A good understanding of the technical problem being solved by program development participants is important. If you must have a development vendor on the team who is not familiar with the technical problem but is on the team because of Artificial Intelligence expertise, it may be desirable to have an outside Consultant on the team to bridge the gap. This was the main criteria HL&P used to select development participant for this project.

3. Hardware specifications and shell selection play an important role in the overall success of the system. The requirements of the technical solution approach must be best satisfied by these selections.

4. A rapid prototype should be developed upon completion of the functional specifications task. This helps to smooth out conflicting requirements and integrate user's desired program characteristics. During the development tasks of the system, the prototype should be frequently updated with increasing functional capabilities. This provides a good tool for the Project Manager to monitor system progress.

5. In situations where there is more than one expert in a technical area, it is likely that disagreements will occur. It is important to allow complete discussions to resolve differences. In most cases, we found that experts had differing assumptions to reach a particular conclusion. Consensus must be reached before programming the heuristic rule base.

6. Testing and validation of completed system is time consuming and should be allowed enough time on project schedule to complete several cycles of updates. A thorough review of system response and acceptance by the experts helps to build user confidence in application of the system. When the experts behind an expert system begin to show confidence in the system, it is most probably a successful system.

ACKNOWLEDGEMENTS

The thought of developing this expert system was inspired by the work of Mr. A. L. Buffinton, Consultant to Houston Lighting & Power Company. His co-authors on this paper acknowledge his important contribution to making this a successful project. We would also like to thank Mr. G. Fred Moore, of Stone & Webster, and the following HL&P personnel for their contributions: E. A. Ott, J. W. Frey, Judith Eisenberg, and J. R. Kessling.

REFERENCES

1. E. Raask. Erosion Wear in Coal Utilization. Hemisphere Publishing Corporation, Washington: 1988.

2. E. Raask. Mineral Impurities in Coal Combustion. Hemisphere Publishing Corporation, Washington: 1985.

3. J. G. Singer, et. al. Combustion, Third Edition. Combustion Engineering, Inc., Windsor, CT: 1981.

4. Babcock & Wilcox Company. Steam, 38th Edition. New York, 1972.

5. EPRI CS-4283, Vol. 2, "Effects of Coal Quality on Power Plant Performance", February 1986.

6. HL&P Operation and Engineering Staff Interviews as recorded 1988, 1989.

7. SWEC Engineering Staff Interviews as recorded 1988, 1989.

Condenser and Feedwater Heater Expert Systems

JOHN L. TSOU, STEPHEN M. GEHL, and S. MURTHY DIVAKARUNI
Electric Power Research Institute
Generation and Storage Division
3412 Hillview Avenue
Palo Alto, California 94304, USA

Abstract

Condenser and feedwater heater expert systems are under development to support the EPRI management initiative of using expert systems to help electric utilities improve plant performance and availability. These expert systems will be used by utility engineers to conduct performance evaluation and monitoring, diagnose the causes of failures, and provide advice on operation and maintenance activities. The condenser expert system will be capable of operating in either an on-line or off-line mode; the feedwater heater expert system will be developed for off-line operation, with the option of adding on-line capability at a later date.

This paper will discuss typical problems of condensers and feedwater heaters and the functional requirements and planned capabilities of these two expert systems in detail. The approach planned for developing the expert systems will also be covered.

Introduction

Problems in steam surface condensers and feedwater heaters cause significant availability losses and heat-rate degradation in nuclear and fossil-fueled power plants. A study [1] conducted by EPRI in 1976 analyzed Edison Electric Institute (EEI) data to determine the availability and performance losses and the annual costs associated with condenser and feedwater heater problems (Tables 1 and 2). At the same time, availability and performance improvement goals were established for condensers and feedwater heater. Subsequent to that study, EPRI conducted a survey to identify and quantify problems of of condensers [2] and feedwater heaters [3]. To help utility engineers solve these problems, shown in Tables 2 and 3, EPRI conducted an extensive research program on condensers and feedwater heaters and published the results in a series of reports, listed in Table 4. Now, utility engineers are faced with the problem of digesting a large body of research results to find solutions to their problems.

Recognizing that the solutions to many plant operating and maintenance problems can be found in the research program results, EPRI has undertaken the development of expert systems on condensers and feedwater heaters. The expert systems will make it easier for utility engineers to access the results they need in daily plant operations and will guide them through the analysis procedures necessary to diagnose and solve problems.

The purpose of this paper is to inform potential users of the planned capabilities of these two expert systems and the technical approach that will be used in their development.

Functional Requirements

The first step in the process of developing the expert systems is the definition of the functional requirements. The expert systems will be designed for use by maintenance engineers, operations engineers, and project engineers. In addition, in the on-line mode, the systems will also be used by plant operators. The expert

Steam Condensers and Cooling Water Systems in Fossil Fuel Units

Losses and EPRI Goals for Improvement

	Losses		Improvement Goal	
		Annual Cost		Savings
	Percent	($M)	Percent	($M)
Availability	2-3	800	1.8	630
Performance	1.3-1.5	320	1.2	270
		1120		900

Source: EPRI FP-422SR
Based on EEI Data

Table 1

Feedwater Heaters in Fossil Fuel Units of 400 MW+

Losses and EPRI Goals for Improvement

	Losses		Improvement Goal	
		Annual Cost		Savings
	Percent	($M)	Percent	($M)
Availability	0.3	57	0.2	40
Performance	1.0	230	0.7	160
		287		200

Source: EPRI FP-422SR
Based on EEI Data

Table 2

Problems Of Condensers

Steam Side	Water Side
Water in-leakage	Corrosion
Air in-leakage	Erosion
Tube-bundle design/layout	Macrofouling
Deaeration performance	Microfouling
Air-removal capacity	Cooling water systems

Table 3

Problems In Feedwater Heaters

Level control and drains subcooling zone
Corrosion
Tube inlet end erosion
Flow induced vibration
Steam impingement
Tube plugging
Design specifications

Table 4

EPRI PUBLICATIONS

CONDENSER

Topic	Number
Failure Cause Analysis	7
Guidelines	18
Workshop Proceedings	6
Other	13
Total	44

FEEDWATER HEATER

Topic	Number
Failure Cause Analysis	2
Guidelines	4
Workshop Proceedings	3
Total	9

Table 5

systems will be small, stand-alone products that will run on inexpensive microcomputers. The systems will be modular in design to facilitate integration with other software products in the future. The software will be designed so that operators and engineers will be able to use the expert systems with a minimum of training.

Additional details on the capabilities of the expert systems are given in the following paragraphs.

1. User configurable for specific unit designs. The expert systems will contain information on a wide variety of equipment designs and configurations. The user will be able to select among such options as multiple pass or dual pressure condensers, dual string feedwater heaters, forward pumped heater drains, etc. The user will be able to enter other design information, for example the number of tubes, tube dimensions, materials of construction, etc. Unit-specific configuration will be performed when the system is installed at a plant; the unit-specific information will be stored for use during subsequent expert system runs.

2. Data validation. The expert system will have the capability to validate all data inputs by checking them against the expected range and comparing with other inputs for mutual consistence. Out-of-range or suspicious inputs will be flagged for the information of the user and will be ignored or discounted, as appropriate, in expert system runs.

3. On-line and off-line performance monitoring and trending. The ES will monitor the performance of the equipment based on either on-line or off-line data. Performance indices will be trended and stored for future use to detect any gradual deterioration in performance. The expert system will have the capability to perform calculations to compare the actual performance to the best achievable for a given set of operating parameters. It will also have the capability to perform calculation to predict the performance consequences of abnormal operating conditions such as isolation of a condenser waterbox, by-passing a string of heaters to check for flow induced vibration, etc. The expert system will

also be able to perform "what if" calculations to evaluate the effects of plugging a significant number of tubes, retubing with different materials, etc.

4. Performance improvement advisor. If the condenser or heater performance is poorer than the best achievable value, the expert system will have the capability to help the engineer find and correct the cause of the deficiency.

5. Diagnostics and failure analysis. The expert system will be able to diagnose equipment failure mechanisms, identify probable root causes, and provide guidance on corrective measures. The system will also recommend inspection methods to help in the diagnosis of problems and help with the interpretation of the inspection result.

6. Operations advisor. The expert system will provide operations advice on topics such as water treatment, layup procedures, etc.

7. Maintenance advisor. The expert system will provide maintenance advice on preventative maintenance procedures, tube cleaning methods, tube plugging methods, etc.

8. EPRIGEMS user interface. The expert system will follow the EPRIGEMS specification [4] for computerized technology transfer. The specification calls for a common user interface, which will facilitate the integration of the heat exchanger expert systems with other EPRI performance monitoring and diagnostic products.

Technical Approach

The architecture of the expert systems will be designed to satisfy the objectives of: (1) on-line and off-line performance monitoring and diagnostic capability; (2) off-line failure diagnostic and advisory capabilities; (3) small, stand-alone system capable of running on low-cost microcomputers; (4) suitability for use on power plants with a variety of steam cycles and heat exchanger designs; (5) suitability for use by utility operating and engineering staff.

The decision to develop relatively small, stand-alone systems for feedwater heater and condenser applications will satisfy the immediate needs of a large segment of the utility population. However, these needs are likely to change as more utilities gain experience and confidence in the use of expert systems. Therefore, the heat exchanger expert systems are being designed to include the future possibility of expansion and integration with a larger package of plant performance and diagnostic expert systems modules [5]. In particular, EPRI is developing an expert system for diagnosing heat rate degradation problems in parallel with the heat exchanger expert systems. The use of the standardized EPRIGEMS user interface for these expert systems will facilitate their eventual integration into a single software product.

The requirement for expert systems with both on-line and off-line capabilities influences both the approach used for system development and the overall expert system architecture. The following discussion makes specific reference to the condenser expert system, but similar comments hold for the feedwater heater expert system.

In the on-line mode of operation, the condenser expert system will have essentially real time access to operating parameters such as condensate flow rate and temperature, condenser back pressure, circulating water temperature, and circulating water flow rate. These parameters will be monitored at short intervals, typically a few minutes. The ability to trend key operating parameters over short times will make the on-line capability of the expert system particularly useful for diagnosing performance problems and providing feedback on needed adjustments in operating conditions.

The expert system will also function in an off-line mode, in which inputs are entered manually. Off-line operation can be used for performance monitoring and diagnosis when the expert system is not interfaced with plant instrumentation. The lower installation cost and reduced instrumentation requirements will make the off-line mode attractive in many such utility applications. However, compared with the on-line mode of operation, off-line operation will have a

longer response time and may be more difficult to use, because of the requirement for manual data entry.

In both on-line and off-line operation, the expert system will validate the manually or automatically entered data, make diagnoses of performance and equipment problems, and make recommendations for operator actions and maintenance tasks to correct problems and prevent their recurrence. The diagnoses and recommendations of the two modes of operation will be similar, although the ready availability of trend information during on-line operation will probably permit somewhat more sophisticated and detailed diagnoses and recommendations. This means that the rule bases for the two modes of operation will differ in some areas.

Many of the features of the condenser expert system not associated with thermal performance will be accessed in the off-line mode of operation. For example, the expert system will be able to prompt the user to perform special tests and inspections, the results of which will be entered manually and used to supplement the data available for diagnoses. The expert system will also be able to process information on equipment design, operating history, maintenance practices, and maintenance and outage schedules. Analysis of this information will allow the expert system to make recommendations on preventive maintenance, repairs and replacements, and inspections.

Once the functional requirements and architecture of the expert system are defined, the knowledge base will be developed. A panel of experts will review the available information from the series of EPRI publications in the condenser area, ASME Performance Test Code 12.2, and other literature sources. The prime contractor and the expert panel will develop sets of heuristic rules for problem diagnosis and the various advisor functions, and will make recommendations on mathematical models and algorithms for calculating performance indices and other parameters. The development of the rule bases for on-line and off-line operation will be carried out in parallel to allow for continual comparison and verification of the rules.

For on-line operation, the expert system must be integrated with plant instrumentation. To do this, interfaces will be provided to tie the expert system host computer to the existing plant computer and plant instruments not connected to the plant computer. The preferred approach will be to run the expert system on a separate microcomputer and use the plant computer only as a source of data. This approach is especially well-suited for older plant computers that have limited processing capability and are frequently difficult to program; in this situation, the only requirement placed on the existing plant computer is that it provide a formatted data stream to the expert system host. However, some utilities may wish to use the plant computer as the expert system host. Adaptability to different host computer configurations will be designed into the expert system to allow the widest possible penetration in the utility market.

During the rule base development and instrumentation integration activities, determinations of the minimum input requirements of the expert system will also be made. Because of the differences in the rule bases, separate determinations will be needed for on-line and off-line operation. Additional work is planned to evaluate the relationships between the type of instrumentation, number of input parameters, and the accuracy of the diagnoses and recommendations.

Extensive validation of the expert system is planned, both by running the system against "simulated" operational data and by in situ testing in a group of utility power plants. The testing and validation effort will cover all the functions of the expert system, including the signal validation routines, the accuracy of performance and failure diagnostics, and the validity of operations and maintenance recommendations. Suggestions from the panel of utility users will be incorporated into subsequent versions of the expert system.

The approach for the development of the feedwater heater expert system will follow a similar path, except that the initial emphasis will be placed on an off-line version. A decision about whether to proceed with an on-line version will be postponed until the value of on-line diagnosis of feedwater heater problems can be assessed.

Conclusions

Condenser and feedwater heater problems cause significant losses in availability and performance. The condenser and feedwater heater expert systems will make it easier for utility engineers to access the large body of EPRI research results they need in daily plant operations and will guide them through the analysis procedures necessary to diagnose and solve problems.

The functional requirements and planned capabilities of the expert systems, as described in this paper, will serve as a guide in developing the systems. EPRI is in the process of negotiating with two contractors to develop, test, and demonstrate these expert systems. Any member utility interested in providing input or hosting a demonstration of the prototype should contact the authors.

References

[1] "Availability of Fossil-Fired Steam Power Plants", Electric Power Research Institute Report FP-422SR, June 1977.

[2] "Failure Cause Analysis - Condenser and Associated Systems", Electric Power Research Institute Report CS-2378, June 1982.

[3] "Failure Cause Analysis - Feedwater Heaters", Electric Power Research Institute Report CS-1776, April 1981.

[4] "EPRIGEMS Product Specifications", Draft, Electric Power Research Institute, November 1988.

[5] Valverde, L. J., et al, "Fossil Power Plant Applications of Expert Systems: An EPRI Perspective", American Power Conference, Chicago, Illinois, April 1989.

Control and Diagnosis of Water Chemistry in the Water-Steam Cycle and Water Make-up of a Fossil Power Plant

M. A. SANZ-BOBI, I. J. PEREZ-ARRIAGA, J. L. SERRANO-CARBAYO, M. E. ORTIZ-ALFARO, J. J. ALBA, A. DOMENECH, and M. J. VILLAMEDIANA
Instituto de Investigación Tecnológica
Universidad Pontificia Comillas
Alberto Aguilera 23, 28015 Madrid, Spain

J. GONZALEZ-HUERTA, J. J. FERNANDEZ-MARTINEZ
Central Térmica de Anllares
Unión Eléctrica Fenosa, S.A.
Ponferrada, 24400 León, Spain

ABSTRACT

This paper presents the expert system SEQA (this name stands for the initials in Spanish of Water Chemistry Expert System) for the control and diagnostic of the chemical properties of the water-steam and water make-up treatment systems in a fossil power plant. The purpose of the system is to monitor the parameters that control the water chemical properties in a power plant, in order to forecast anomalies and, if they occur, to diagnose the problem that causes them and recommend corrective or preventive strategies.

INTRODUCTION

Corrosion and pollution phenomena, closely related to the water chemical properties, are among the main causes of unavailabilities in power plants steam generators and turbines. Until the 60's these incidents were very difficult to prevent due to the lack of useful knowledge about high pressure water chemistry.

In the 70's, an intensive research effort was carried out in order to identify the phenomena that originate these damages, to develop efficient supervision and control

methods and to apply this knowledge to plants operation. The efficient management of physical and chemical parameters made possible to discover incipient failures and led to an important increase of the equipments availability and useful life.

The different processes that take place in a power plant are closely related. Mere supervision of the chemical parameters to check that they are not out of security levels is not enough to detect and prevent potential failures. Relationships among the different measurements and analyses, very difficult to represent in an accurate mathematical model, have to be taken into account. Therefore the experience of Chemistry experts is needed in order to perform a good supervision. These characteristics seem to be appropriate for the application of an expert system.

The Instituto de Investigación Tecnológica (IIT) has developed and implemented an operational prototype of an expert system (SEQA) for control and diagnosis of water chemistry at the Anllares Power Plant (350 Mw). Unión Eléctrica Fenosa (UEFSA), the utility that operates the plant, technically supported the project, and OCIDE (the Office for Coordination of Power Industry Research) provided the necessary funding. The knowledge was mainly provided by Anllares chemical staff and the whole project was supervised by a task group of chemistry experts, representing the main Spanish electrical utilities.

SEQA is intended to become a very useful aid for the Chemistry laboratory staff, because it stores experience and knowledge related to incidents that may have not previously occurred at the power plant where it is implemented. SEQA will be also a very useful aid for the operation staff, that is usually less acquainted with the significance and consequences of chemical parameters. The system can be specially useful in night shifts, when there is usually a lack of chemical experts in the plant.

This paper will briefly present an overview of the water chemical problems in a power plant, the current ways of dealing with them and the main features of SEQA as an important aid for this task.

THE WATER CHEMISTRY IN A POWER PLANT

Any low concentration impurity present in the water-steam cycle usually keeps in the water and does not pass to the steam in a dangerous level. However, the contaminants will be quickly accumulated in the boiler, due to the large flows of steam required by the current turbines (for instance, approximately 1100 tm/h in the Anllares power plant) and the operation in closed loop. When contamination in the boiler reaches a certain level, impurities can contaminate the steam and affect the turbine.

The deposition of polluting substances decreases the efficiency of equipments and induces corrosion through a wide variety of mechanisms. The water chemical characteristics in a power plant can be affected by extraneous impurities such as:

- Salts that can be present in the cooling water and can contaminate the water-steam cycle because of leaks in the condenser.
- Salts coming from the water makeup plant due to a bad operation.
- Oxygen and carbon dioxide due to air inputs in low pressure zones.

There are several chemical treatments used to prevent corrosion phenomena. An example of them is the one used in Anllares power plant, that consists of the addition of:

- Hydrazine. Its goal is the elimination of the oxygen from the water. The hydrazine is decomposed into ammonia by thermal causes.
- Ammonia. The water is less corrosive for steel in alkaline conditions (pH approximately 9). Ammonia is added into water in order to keep it in the aforementioned conditions, when the ammonia produced by the hydrazine decomposition is not enough.

Contamination phenomena can be detected and controlled by the measurement of chemical parameters. These measures are performed by chemical analyzers (see reference 1) located at different points in the water-steam circuit. The more usually controlled magnitudes are:

- Specific conductivity, that provides information about levels of existent ammonia.
- Cathionic or acid conductivity. It is an indirect measurement of the quantity of salts existing in the water, and therefore it gives an idea of water impurity.
- Oxygen concentration.
- Hydrazine concentration.
- Hydrogen concentration, that is related to the corrosion on steel.

- pH, that indicates the acid or alkaline character of water.

Manual analyses of different salts are also carried out, because they can indicate other contamination problems, such as the presence of sodium, chlorides and so on.

The aforementioned measures can be taken in different sampling points along the water-steam cycle. In Anllares power plant the selected points are the following:

- Cathionic and specific conductivity. They are measured at the condensed discharge pumps, feedwater, steams and boiler.
- Oxygen concentration. It is measured at the condensed discharge pumps and the feedwater.
- Hydrazine, at feedwater.
- pH. is available at the condensed discharge pumps, deaerator output and boiler.
- Hydrogen at the superheated and reheated steams.
- Thickness and conductivity in the water makeup treatment plant.

Non-chemical signals that are required for checking anomalies, such as the generated power, cooling water input and output temperatures or vacuum pressure of the condenser are also received. Finally, manual analyses of Na, Cl and organic matter are also carried out, at least once in each shift.

JUSTIFICATION AND GOALS OF SEQA

The power plant Chemistry laboratory staff monitores the data coming from automatic analyzers, carries out manual analyses and selects preventive or corrective actions according to the obtained measures. This task is mainly based on chemical knowledge and experience about past incidences.

The selected actions can critically influence the plant life and production. Therefore, a solid background of chemical experience and well established knowledge would undoubtedly improve the plant operation. The knowledge-based systems technology permits the development of computer systems containing all this experience and knowledge.

The main missions of SEQA are to monitor the performance of the chemical parameters in order to forecast incipient anomalies, to diagnose trouble situations, and to act as a

consultant about possible actions to take. Three main elements are required in order to reach these goals:

- The chemical knowledge and experience related to a wide number of power plants.
- Statistical support tools.
- Chemical data taken from the power plant.

The SEQA system will become:

- An efficient and exhaustive supervisor of chemistry in a power plant.
- An aid system for the laboratory and operation staff in the plant.
- A way of representing and storing chemical knowledge and experience that can be easily implemented in other power plants.
- A large database of issued diagnoses about occurred operation incidents that can be installed in other power plants with similar configuration, increasing the available experience.

OVERVIEW OF THE SEQA SYSTEM

A list of SEQA modules (see fig. 1) follows:

1. *Automatic data acquisition module.* It is in charge of acquiring on-line data from the chemical analyzers and other physical-chemical parameters that will be used as input to SEQA Its main element is a standard datalogger.
2. *Anomalous condition detector module.* Its task is to monitor the acquired data and to test if the values correspond to normal operation conditions. If this is not true, the expert module is invoked.
3. *Database of automatic chemical parameters.* It stores historical information about the on-line chemical parameters coming from the automatic analyzers.
4. *Database of manual chemical analyses.* It contains historical information coming from the manual analyses performed in the laboratory.
5. *Database of power plant configuration.* It stores the description of the power plant from a chemical viewpoint.
6. *Expert module.* It is the main element of SEQA. It is called by the anomalous condition detector module, when a dangerous chemical situation is detected. Then it uses its knowledge base and starts the inference process. As result, one or several diagnoses with their corresponding corrective actions are issued.

7. *Mathematical module.* It contains a set of statistical functions. These can be used by the expert module and by the chemical laboratory personnel to analyze historical data.

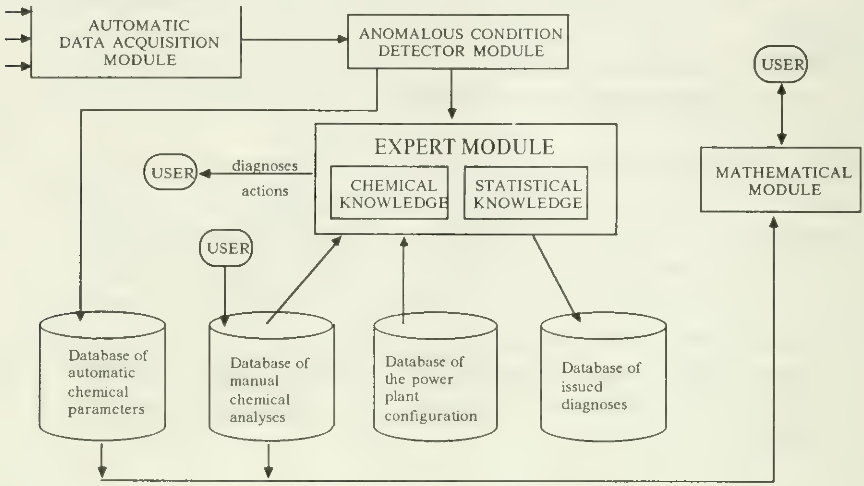


Figure 1. Overall structure of SEQA

8. *Database of issued diagnoses.* It collects historical information on the different diagnoses issued by SEQA and the anomalous conditions previously occurred.

During normal operation, SEQA shows a screen where the last values coming from the datalogger are presented. These values are refreshed with a period of 60 seconds, and they are also introduced into the anomalous situation detector. If no chemical anomaly is detected, the refreshing and processing continues in a loop. The mathematical module can be used in an off-line mode, in this case the datalogger takes care of the data storage.

The database of automatic chemical parameters is updated every 15 minutes, with a single average value taken over 15 samples. The identifiers of the sampling points are shown, followed by the values of the magnitudes. This information can also be presented through graphic screens, where the data are arranged around a pictorial representation of the cycle and water make-up plant components.

If a chemical anomaly is detected the expert module is invoked, and SEQA presents another screen type. The control of the automatic data storage is temporarily transferred to the datalogger, and SEQA expert module starts the diagnostic process.

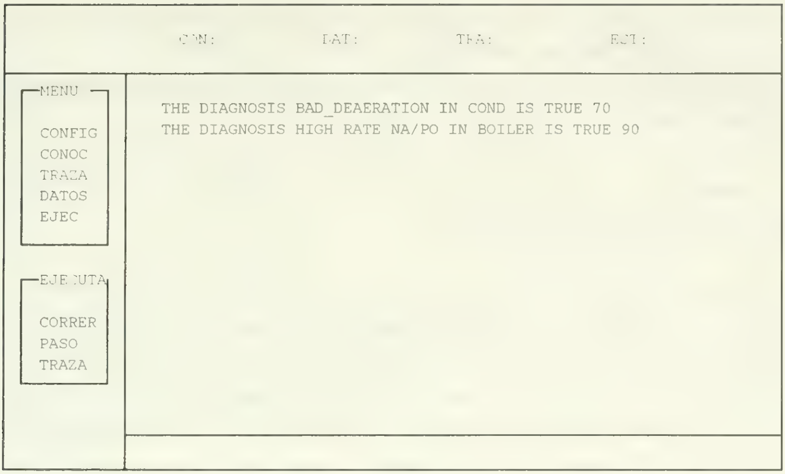


Figure 2. A sample screen of SEQA expert module.

An expert system sample screen is shown in figure 2. It is divided into four areas. The upper one contains general information about the system status: existence of pending analyses, trace on/off, etc. The bottom rectangle in the screen outputs some information such as issued warnings and requests for data. The left zone presents the menus that can be used to start

the statistical module or to consult the knowledge base, the issued diagnoses list, the requested analyses list, etc. At last, the bigger central area in the screen is the main output window, and shows diagnoses, configuration information, etc.

The detected anomalous data are used to update the facts database. The relevant rules are then selected and their conditions are checked. The system can fire statistical computations or ask for manual analyses when this is required by any of the rule antecedents. The process of applying rules will be running until the chemical normality is reached. Issued diagnoses and important data will be stored in the database of diagnoses.

Once the expert module finishes its work, the personal computer retrieve the new data that had been stored in the datalogger internal memory while the expert module was running, and SEQA goes back to normal operation.

SEQA has been designed to operate without the interaction of the user. The expert module is automatically invoked when needed, and the results of the inference process (diagnoses, actions, etc.) are stored in lists that can be easily consulted by the user at any time, through the aforementioned menus.

Automatic data acquisition module

This module consists of the hardware and software required to perform the A/D conversion. Analog signals are mainly provided by chemical analyzers installed at several sampling points in the plant. Digital signals are introduced in a personal computer, becoming the input for the rest of the SEQA modules. Currently, 43 signals are being sampled in the first prototype implemented in Anllares Power Plant, and the sampling period is 60 seconds.

The main element of this module is a datalogger, the memory of which is able to store data corresponding to about 3 hours of sampling. The communication between the datalogger and the personal computer is carried out through a RS-232 interface, controlled by a communications software written in C language.

Anomalous condition detector module.

This module takes care of requesting the data stored in the datalogger memory and invoking the expert module when an anomaly is detected in some parameter. The detection is performed through two filtering routines that have the following functions:

1. Comparison with security levels, that have been proposed in reference 2 and accepted by a task group of chemistry experts.
2. Detection of abnormal trends and comparison with normal operation conditions in the plants using quality control techniques (see reference 3).

Database of automatic chemical parameters

This database stores the historical data submitted by the anomalous condition detector module. Average values for every 15 minutes are stored. This storage frequency is considered to be sufficient to accurately reproduce the historical evolution of chemical parameters, and permits to minimize the storage requirements of the database.

Database of manual chemical analyses

It contains the results of the daily water analysis performed by the plant laboratory analysts. These data are processed in the same way as the automatic data. Currently, the aforementioned data come from the water make-up treatment analysis, the water-steam cycle analysis and the external water analysis of the power plant. During the diagnostic process, manual analyses can also be requested by the expert system, in order to confirm hypothesis.

Database of power plant configuration

A very interesting feature of SEQA is the possibility of applying it to any fossil-fueled power plant. All modules of SEQA are independent from a particular structure of power plant, except this configuration database.

This database permits to introduce the particular characteristics of a power plant to the expert module. This information is used, for instance, to select the part of the knowledge base that can be applied to this particular power plant, disregarding the knowledge elements that are related to different configurations.

The configuration database is created by the user during SEQA installation. It is written in semi-natural language and contains the general plant features (such as boiler type, fuel type, cooling type and chemical treatment) and information about the sampling points. At every sampling point, the following items are specified:

- Sampling point name (abbreviated and extended name).
- Main material of the physical component where the sampling point is located.
- Automatic chemical measurements at the sampling point, indicating measure name and datalogger channel number.

- Manual chemical measurements at the sampling point.
- Special mathematical function to apply at the sampling point.

Expert module

The expert module is the main element of SEQA. It takes the role of a chemical expert facing an abnormal chemical situation. SEQA expert module is invoked by the anomalous condition detector module when some signals coming from some analyzers reach a dangerous or potentially dangerous level. The system notes these new facts, includes them in its description of the plant and starts an inference process that will hopefully lead to an explanation of what is happening and a list of corrective actions.

According to these ideas, SEQA expert module has the following elements:

- *Facts database*. It represents the plant configuration, the variables that are being monitored, and their current values (including the abnormal ones).
- *Knowledge base*. Its content simulates the knowledge used by a human chemical expert when she (or he) is presented with a chemical problem in a power plant.
- *Inference engine*. It is the mechanism in charge of applying the knowledge in the knowledge base to the facts database, in order to build the reasoning process to produce the diagnosis.
- *Lexicographical analyzer*. It takes care of translating input files (configuration database, rules, etc.) into their Lisp representation.

Given the importance of this module, it will be described in more detail in a following section of this paper.

Mathematical module

This module contains a set of several statistical functions [4]. It is used in two ways:

- On-line use by the expert module during its diagnostic work.
- Off-line use, to help the chemist expert on the study of chemical parameters evolution.

Several statistical functions and combinations of them are included in this module in order to compute means, variances, correlations, regression analyses and time series analyses (self-correlation and partial correlation are currently implemented). Graphics generation for SEQA is also included in this module.

Database of issued diagnoses

This database stores the history of the diagnoses issued by SEQA, and the qualitative values that led to these conclusions. It can be used by the chemistry experts to upgrade the existing knowledge base or to infer new rules through the analysis of past situations and results. The possibility of automatically inferring new rules, using associative memory techniques, will be considered in a future version of SEQA.

SEQA EXPERT MODULE

Figure 4 shows the overall structure of the expert module. The four main elements of this component will be now presented.

Facts database

The structure of the facts database is generated from the data stored in the power plant configuration database when SEQA is installed in a power plant.
The facts database stores static and dynamic information. This information is composed of facts relevant to the whole plant and to the different components and sampling points.

Static information. Static facts mainly describe the power plant configuration. They are the underlying framework for the dynamic information. Static information describes the different cycle components, the available measures at each of them, the existing relationships between components and other plant features.

Dynamic information. Dynamic facts comprise the values taken by the several chemical measurements, the issued diagnoses and the rules that can be applied.

Knowledge base

It is composed of decision rules that permit the inference of diagnoses and recommended actions from specific chemical events in the power plant. This knowledge has been provided by the Anllares plant staff and a task group of chemistry experts of power plants, representing the main Spanish electrical utilities.

According to these experts, the different sampling points have been selected as basic units for chemical malfunctions location and investigation. Therefore, the rules are referred to the sampling points and make use of conditions about one sampling point chemical status or about its relationships with the previous one.

The general structure of a rule is the following:

```
IF condition THEN conclusion
```

Condition and *conclusion* are sets of patterns that describe a given configuration of the facts database. These patterns can be joined by AND relationships. A condition pattern is satisfied when a fact in the facts database can successfully be matched with the pattern. In that case the pattern is said to be instantiated by the fact. When all the patterns in the condition side have been instantiated the rule can be applied, i.e. it can be fired.

SEQA has five types of decision rules: specific rules for each sampling point, general chemical rules, systematic application rules, statistical rules and rules about impact of water conditions in the power plant components. All these rule types are stored in the same knowledge base. Currently SEQA contains about 140 rules.

Specific rules for each sampling point. These rules are the most important pieces of knowledge in the knowledge base. They associate a certain set of chemical parameters values in one or several sampling points with a given diagnosis. These rules are well suited for the discovery of well or nearly well defined chemical problems.

An example of this category of rules is the following:

```
IF   the measure of cathionic conductivity at the sampling point in
     the condensate discharge pump is high
AND  the measure of cathionic conductivity at the sampling point in
     saturated steam is high
AND  the measure of cathionic conductivity at the sampling point in
     superheated steam is high
```

```
AND  the measure of cathionic conductivity at the sampling point in
     reheated steam is high
AND  the measure of sodium at the sampling point in the condensate
     discharge pump is high
THEN the diagnosis carry-over at the boiler is true 100
```

Where 100 is the certainty factor of this diagnosis.

__General chemical rules__. Sometimes, the picture of chemical parameters do not correspond to an accurate description of a known chemical problem. The symptoms of the failure are fuzzy. In this case, the specific rules for each sampling point do not lead to a solution because they can detect a problem only when it fits into the description stated in the conditions side of the rule. However, the general chemical rules can provide an explanation according to more general chemical knowledge. The following is an example of this kind of rules.

```
IF   the measure of oxygen at the ?xc sampling point is high
THEN the diagnosis air inputs at ?xc is true 80
AND  the diagnosis bad deaeration in ?xc is true 40
AND  the measure of hydrazine at the sampling point ?xc is low.
```

In this example, the symbol ?xc represents any sampling point in the water-steam cycle. There are other similar symbols, such as ?xp, representing any sampling point in the water make-up treatment and the ?x representing any sampling point in both sampling point sets.

__Systematic application rules__. They are used to discover the true origin of a chemical anomaly. The physical components of a power plant are sequentially connected, and the fluid (water or steam) is the same throughout this sequence. Therefore, a chemical anomaly detected at a sampling point can be transmitted along the cycle. Because of this reason, it is necessary to identify where the chemical anomaly actually starts, taking into account the possibility of simultaneous chemical problems. These rules serve to reach this goal. An example of them can be the following:

```
IF   the measure of ?y at the ?xc sampling point is high
AND  the measure of ?y at the preceding ?xc sampling point is normal
THEN there is a problem in the ?xc sampling point.
```

The symbol ?y represents any chemical parameter.

Statistical rules. They will be used to complement chemical knowledge and to infer potential chemical malfunctions. Statistical parameters are used in the rest of the rules, but purely statistical rules are not yet included in SEQA. They will be introduced in a next version of SEQA, in order to improve the interface between the expert and the mathematical modules.

Rules about impact of water conditions in the power plant components. They try to collect the knowledge about corrosion effects of a chemical situation on different materials. This knowledge is very difficult to collect. However, these rules are a first research step on the status of the different materials of the power plant components after a given chemical situations. Therefore, these rules are a first draft and are not the main rules in SEQA.

One of these rules is the following:

```
IF   the measure of ammonia at the ?xc sampling point is high
AND  the measure of oxygen at the ?xc sampling point is normal
AND  the measure of cathionic conductivity at the ?xc sampling point
     is normal
THEN the corrosion of the admiralty at the ?xc sampling point  is 3
AND  the corrosion of the alloyed steel at the ?xc sampling point is
     -1
```

Inference Engine

The inference engine is in charge of governing the application of the rules contained in the knowledge base according to the information in the facts database. It uses simple pattern matching techniques to identify facts that instantiate the patterns in the condition side of the rule. The tasks performed by the inference engine are organized as follows:

1. During the installation of SEQA in a given plant, the system identifies all the rules related to each one of the components and measures in the configuration database. Pointers are then set to link components and magnitudes to rules. Therefore, when an abnormal value for any of the magnitudes is detected, the affected rules can be quickly selected.
2. The expert module is invoked due to the action of the anomalous conditions detector (an abnormal value of a measure for any of the chemical parameters is present). At this point all the relevant rules (the ones that have the abnormal

measure among their conditions) are selected using the aforementioned pointers. <u>These rules are introduced into the agenda</u>.

3. The agenda will contain rules pertaining to the aforementioned five types of rules. <u>Systematic rules will be first used in order to identify the problematic sampling point</u>.
4. <u>When the problematic sampling point (or points) has been identified, the inference engine tries to apply the specific rules corresponding to it.</u> Different diagnoses can be issued, with their respective certainties. When several inference lines lead to the same diagnosis, the highest certainty is selected. This is a very conservative approximate reasoning modelling that will be deeply modified in a future version of the system.
5. <u>If the application of specific rules is not possible or the diagnoses certainties are low, then general chemical rules are applied</u>. Statistical rules can be used to complement the application of chemical knowledge.
6. When the aforementioned process is finished, <u>the diagnoses with their associated corrective or preventive actions are presented to the user</u>. Corrosion information, provided by the action of corrosion rules, is also supplied. The incident is closed when the chemical situation is again normal.

<u>Lexicographical analyzer</u>

The rules and the frame-like structures corresponding to the different elements in the plant can be introduced in semi-natural language (using the aforementioned patterns) in a text file. The lexicographical analyzer takes care of translating this information provided to SEQA by the user into Lisp language structures. Therefore, rules can be written as shown in the previous examples, and components can be introduced in a similar way. The analyzer works as a small and simple compiler.

SEQA IMPLEMENTATION REQUIREMENTS

SEQA is designed for an easy implementation at any plant, taking into account all the possible characteristics of Spanish fossil-fueled power plants.

Hardware requirements

SEQA hardware requirements are a datalogger and a personal computer. The datalogger requirements are the following:

- Easy communication through a RS-232 interface with a personal computer.
- The signal conversion must result in ASCII characters, in order to ease the transmission to the computer.
- Memory for temporary storage of the acquired data, during the expert module work.

The first prototype installed at the Anllares Power Plant uses two dataloggers. Each of them collects up to 23 differential signals and has 20 kbytes of RAM memory.

The personal computer requirements depend on the programming languages needs. There are several possible configurations supported by SEQA. For instance, the prototype is currently implemented on an IBM PS/2 Model 60 computer, with 5 Mbytes of extended memory. This computer is fully dedicated to SEQA. These requirements will be considerably reduced when a run-time delivery module will be available for the Lisp compiler.

Software requirements

SEQA is written in two programming languages: C and Lisp. The expert module has been developed using different versions of Golden Common Lisp (registered trademark of Gold Hill Computers, Inc.), the rest of the system modules have been built using Turbo C (registered trademark of Borland Corp.) and Microsoft C (registered trademark of Microsoft). However, these features will not impose any restrictions on the hardware, because SEQA final version will be installed as executable programs.

SEQA installation

System installation requires only a description of the power plant relevant elements and the list of chemical measurements. This information is introduced via the database of power plant configuration.

RESULTS

A complete operational prototype of SEQA has only recently been installed at Anllares Power Plant. The initial installation and data acquisition problems are currently being fixed, and operation experience is still scarce.

The expert module has been installed in personal computers in the different plants that are involved in the project supervision. Very encouraging and useful feed back has been provided by the chemistry experts.

SUMMARY AND CONCLUSIONS

In this paper an expert system for the water chemical properties control in a fossil power plant (SEQA) has been presented. Important SEQA features are the following:

- Fast decision-making; diagnoses are issued in real time.
- The system can be an important help, from the chemical viewpoint. SEQA allows forecasting of potential chemical anomalies in the power plant water cycle, confirming the hypothesis of chemist experts, helping the research work of chemical phenomena and suggesting actions to be taken in an orderly manner.
- From the operation viewpoint, SEQA is a useful tool to advise the operators in the presence of a chemical problem, and particularly valuable when the laboratory staff is not available at the power plant.
- SEQA facilitates the interchange of experience, and diagnostic histories between different power plants.
- The system is open and can be easily adapted to particular configurations. New parameters, rules or strategies can be added

SEQA opens new research lines on the water chemical phenomena in power plants, and helps to obtain a detailed understanding of several chemical techniques currently used. This is possible because SEQA stores useful information about how the chemical

anomalies evolve, the effects that they can produce at the power plant and how the diagnoses have been issued.

The success of the SEQA project has lead to the planning of a possible extension of this philosophy to the whole plant operation. This will allow the monitoring and forecasting of the plant conditions for better operation and maintenance.

ACKNOWLEDGMENTS

Special thanks to the working group of chemistry experts that has supervised the project, and to the instrumentation staff of Anllares Power Plant.

REFERENCES

1. EPRI, Guideline Manual on Instrumentation and Control for Fossil Plant Cycle Chemistry. EPRI report CS-5164, April 1987.
2. EPRI, Interim Consensus Guidelines on Fossil Plant Cycle Chemistry. EPRI report CS-4629, June 1986.
3. D. C. Montgomery, Introduction to Statistical Quality Control, New York, John Wiley and Sons, 1985.

Rotating Machinery Diagnosis Using Knowledge-based Systems

IGNACIA DEL ANGEL, JUAN J. RIVERA, EDGAR N. SANCHEZ, JOSE M. FRANCO, EDMUNDO RIOS, and EDUARDO PRECIADO
Departamento de Equipos Mecanicos
Instituto de Investigaciones Electricas
Cuernavaca, Morelos C.P. 62001, Mexico

ABSTRACT

This paper presents the development of a Knowledge-Based System for impending failures diagnosis of rotating machinery. The different steps to build this system were: selection of a commercially available developing shell, Knowledge-Based Systems synthesis and implementation with knowledge extracted from three different sources, which is the main contribution of the paper, and the implementation of menu-driven user interfaces.

INTRODUCTION

Analysis of vibration signals is a well-known technique to diagnose impending failures in rotating machinery. Usually, the diagnosis is done based on look-up tables or on algorithms. These tables and algorithms relate patterns of the vibration signal to possible causes for the presence of a specific impending failure. The relations to determine different failures have been derived from the experience of human experts on rotating machinery troubleshooting.

To detect the presence of impending failures and to diagnose their possible causes, a human expert usually utilizes the above mentioned tables and/or algorithms as well as his own previous experience. Unfortunately, it is not an easy task to find this kind of experts, mainly in a developing country as México.

Knowledge-Based Systems can help to cope with this situation. They provide the mechanisms to extract the knowledge from human experts and to code this knowledge in a computerized system to be exploited in future applications by personnel with less experience.

Since 1986, The Mechanical Equipment Department (Departamento de Equipos Mecánicos) of the Electrical Research Institute (Instituto de Investigaciones Eléctricas) has been developing, under contract with the Mexican Electricity Federal Comission (Comisión Federal de Electricidad), a computerized system for predictive maintenance of fossil fuel electric power plants. As a very important component of this system, a Knowledge-Based System for the diagnosis of rotating machinery impending failures is being developed.

The paper presents the different steps followed to build the aforementioned Knowledge-Based System. It describes:

How a commercially available Knowledge-Based System developing shell was selected,

The construction of Knowledge-Based Systems based on knowledge extracted from three different sources, as well as the analysis of the results obtained with these systems. This is the main contribution of the paper.

DEVELOPING TOOL

The tool used to implement the Knowledge-Based System was the commercially available developing shell "EXPERT SYSTEM" (EXSYS) 1. This package is written in "C". It provides forward and backward chaining methods and represents the knowledge with IF THEN production rules.

Some of the main characteristics of EXSYS are: a command language that permits to control the execution of programs that use knowledge bases, four fixed uncertainty ranges or the option to create user-defined ranges, control of external programs written in any programming language with returning of EXSYS commands from these programs, and direct access to DBASE III data files.

Systems developed with EXSYS ask the user questions related to a specific subject. The user replies by choosing one or more answers from a menu or by introducing data. The system will keep on asking until it reaches a conclusion. The conclusion could be selection of a single solution or a list of possible solutions ordered by decreasing probability of ocurrence values. The system will show, at user's requests, how the conclusion was reached displaying the rules used.

ANALYSIS AND IMPLEMENTATION OF THE KNOWLEDGE SOURCES

This section presents the sources of knowledge used in the implementation of the system. These sources were analysed in order to know what type of information they contain; how they are structured; and how they could be used. The knowledge sources are:

Diagnostic Charts

A model-Based Algorithm

A Heuristic Algorithm

For the three sources, the operational signals, considered the most relevant to establish a diagnostic, are vibration signals. However, other parameters are also considered, e.g.: noise levels; bearing and coil overheating in an electric motor; and pressure failure of lubrication oil, among many others.

Diagnostic Charts

The four diagnostic charts developed by John S. Sohre 2, an expert in the diagnosis of vibration problems in rotating machinery, are a very important source of information to identify problems in high-speed turbomachinery. Two of the charts relate thirty one causes of vibration and eleven types of problems (identified by the vertical headings) to eighty eight symptoms of operational evidence (identified by the horizontal headings), through relative probability values; these values represent the percentage of cases that show a particular

symptom (Fig. 1). The other two charts relate the same causes of vibration or type of problems to fifty seven symptoms of possible causes of difficulty, through check marks (Fig. 2).

The key to use the charts is the set of vibration signal charactheristics: predominant frequency, direction and location of predominant amplitude, and amplitude changes response to operational speed variations. In addition, chart 4 presents a brief description and suggestions to correct each of the causes of vibration and type of problem.

The objective of implementing the diagnostic charts with EXSYS was to determine how feasible it would be to represent properly this information; also, it was expected to have a tool that would allow to evaluate the results produced by the knowledge-based system.

The partial implementation contains one hundred three rules derived from the information related to vibration analysis and operational evidence sections; according to the values of the charts, an uncertainty range of -100 to +100 was selected; to control the execution, forward and backward chaining methods were used.

The main implementation problem is the lack of discrimination among the results obtained; this lack of discrimination brings out that the number of conclusions obtained by the system, sometimes, is so big that it is difficult to clearly perceive the major cause of vibration or the type of problem to be corrected. On the other hand, there is symbolic information (check marks) in the charts that cannot be directly represented in EXSYS. As a solution to this, it was decided to represent them by numerical values (60%) into the uncertainty range. This approach modifies the final certainty values; so, it is still necessary to validate this representation.

Model-Based Algorithm

The models-based algorithm, developed by Janusz Kubiak et al 3 , is focused to the diagnosis of the failures which produce abnormal vibratory behavior of turbogenerators.

The diagnostic procedure is based on the signal vibration analysis, associating the abnormal vibratory behavior of the machine to its possible causes of failure. The information is grouped in four main categories:

Vibratory characterization of the rotor-bearing system, including: critical speeds, damped natural frequencies, unbalance response, etc.

Vibratory behavior of the rotor-bearing system subjected to different simulated failures.

Historical vibratory trend of the machine.

Data of vibration signal characteristics: amplitudes, frequency, phase, waveform, direction and orientation of the measure, etc.

Using the aforementioned information and the analysis and comparison procedure presented in the algorithm (Fig. 3), it is possible to identify fifteen types of failures, e. g.:

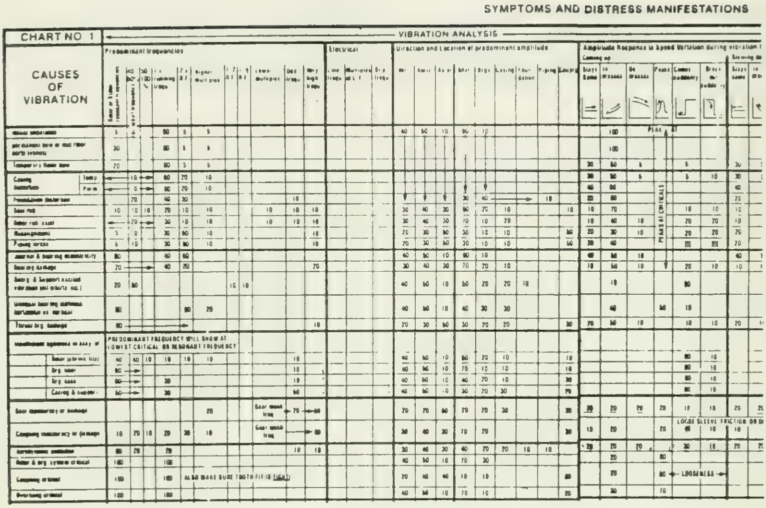


Fig. 1. Diagnosis chart with numerical values

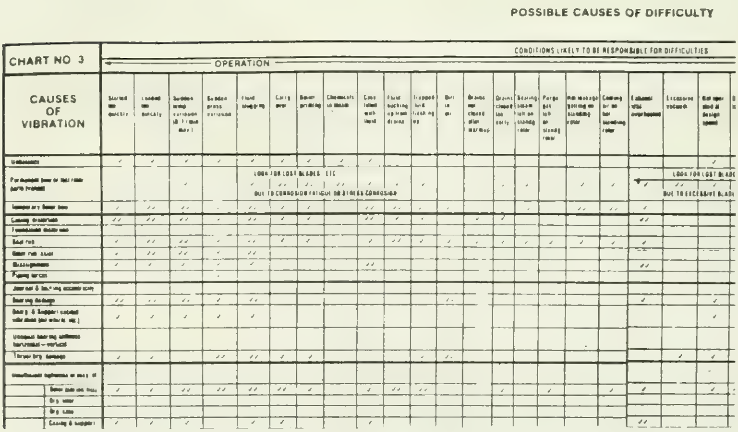


Fig. 2. Diagnosis chart with check marks

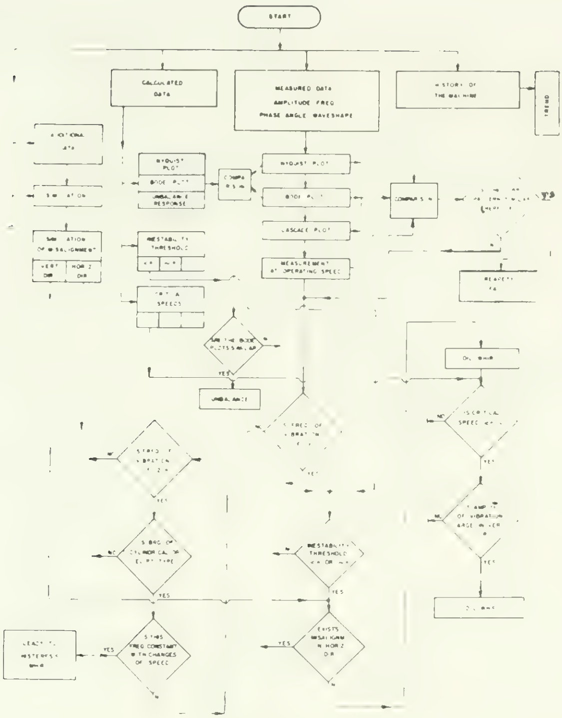


Fig. 3. A part of the modified algorithm for identification of the faults at idle running of the turbine-generator

Rotor-bearing system instabilities

Unbalance

Misalingment

Rubs

Rotor bow...

The main objectives of implementing the algorithm with EXSYS were to evaluate the tools required to accomplish such implementation and to determine if the information presented in the algorithm can be appropriately represented. The algorithm control structure is based on questions which only aceept YES/NO answers.

The partial implementation includes fifteen rules developed taking as a basis data of vibration signal characteristics. An 0-10 uncertainty range was selected because it allows to obtain the same results as those of the original algorithm. The forward chaining method was used, due to the structure of the algorithm. The interface with the user is menu-driven; depending on his/her answers, the system will keep on asking for more data. Once all the requested information have been entered, the system will display the diagnosis.

From this partical implementation, it can be deduced that:

To make a complete implementation of this algorithm it is necessary to have information related to the rotor-bearing mechanical system, such as: critical speeds, natural frequencies, resonances, thermal expansion, etc. Much of this information is unavailable at the present time; its acquisition would require a big effort of field measurements.

The implementation would imply the use of tools which allow the execution of simulation programs (deep knowledge), and the access to data-base and other types of files. These tools are provided by EXSYS.

Heuristic Algorithm

The heuristic algorithm, developed by Asgar Ali-Husein 4 ,is intended to diagnose failures in three types of rotating equipment:

Electric motors

Centrifugal pumps

Belt-driven equipment

The diagnosis procedure is based on vibration signals analysis, as well as on some others indicative parameters of the whole condition of the equipment; such as: temperatures, pressures, flow rate, etc. Throughout the algorithm, depending on the answer given to each of the questions, numerical values are assigned to the probability of ocurrence of different failures. These values are added or subtracted, and finally assigne a certainty degree to the diagnosis. In some cases, the probabilities are initialized to a defined value (Fig. 4).

To develop the partial implementation with EXSYS, the addition/subtraction uncertainty method andforward chaining were used. Fifty two rules were derived from the algorithm; they include only information related to the centrifugal pumps section.

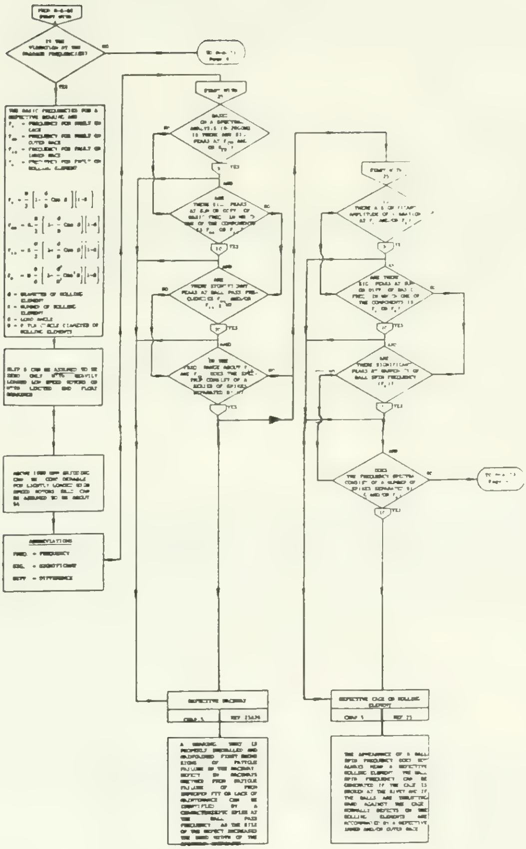


Fig. 4. A part of the electric motor algorithm for identification of the faults at roller bearings

The interface with the usder is through menus. According to the answers, the system will keep on asking for more data, until it reaches a conclusion. The results generated with this system presents the failure diagnosis, indicating its certainty percentage; it also presents, for each of the failures, the rules that lead to the conclusion and additonal comments about the failure.

During the implementation, it was observed that the electric motor diagnosis algorithm actually consists of a number of subalgorithms:

Electric problems

Mechanical problems

Electric problem vs. mechanic problem discrimination

Bearing lubrication

It was also observed that diagnosis algorithms for other rotating machines (centrifugal pumps and belt-driven equipment) have the same structure. When the type of problem is the same, the algorithm uses a unique diagnosis module for any machine; reducing the memory size requirement, and simplifying the implementation and the execution of the knowledge-Based System.

As an example of the knowledge-Based System developed, the operation of the system based on diagnostic charts is presented in Appendix A.

CONCLUSIONS

From the experience obtained during the development of the systems, it was determined that the structure of the final system should be modular like the one presented in 4 . It was also defined that the knowledge base will be extracted mainly from 3, 4 and from new references and experimental data. At the present time, the modular version is being developed; the first type of equipment to be considered is centrifugal pumps. The corresponding system will be tested in a laboratory installation.

ACKNOWLEDGEMENTS

The authors thank the support of the Mexican Electricity Federal Comission (Comisión Federal de Electricidad), inparticular: Mr. José M. Muñoz, Dynamical Analysis office Head, and Mr. Carlos Echeverria, Maintenance Manager of Manzanillo I Fossil Fuel Power Plant.

REFERENCES

1 Expert System Development Package Manual, EXSYS, INC., Alburquerque, New Mexico, 1988.

2 Sohre, John S., Turbomachinery Problems and Their Correction, Sawyer's Turbomachinery Maintenance Handbook, Chap. Seven, Edited by John W. Sawyer and Kurt Hallberg, Turbomachinery Publications, Norwolk, CT, 1980.

3 Janusk Kubiak, Andrés Rothhirsch and Jorge Aguirre, An Algorithm of Fault Diagnosis for Turbine-Generator Operations, Instituto de Investigaciones Electricas, Cuernavaca, México.

4 Asgar Ali-Husein, Condition Monitoring By Vibration Analysis Using Diagnostic Charts. M.Sc. Thesis, School of Mechanical Engineering, Cranfield Institute of Technology, 1987, Cranfield, England.

APPENDIX A

OPERATION OF THE KNOWLEDGE-BASED SYSTEM BASED ON THE DIAGNOSTIC CHARTS

This example shows how the developed Knowledge-Based System operates. The example is particular to the system based on diagnostic charts, but the three of them work in a similar way.

The system requests information through menus; the options are related to vibration characteristic and/or operational evidence.

To start the diagnosis, the first questions the system asks are about available information. The system displays the following menu (the menus are presented in Spanish, as in the original system, but an English translation is included in parenthesis):

LOS DATOS DE ENTRADA	(Input data are)
1 SON DE ANALISIS DE VIBRACION	(from vibration analysis)
2 SON DE EVIDENCIA OPERACIONAL	(from operational evidence)
3 NO ESTAN DISPONIBLES	(Not available)

Once the user has answered, the system will keep on asking more input according to the option chosen. If we assume the user has chosen option 1, then the system will display the following menu:

LOS DATOS DE ENTRADA REFERENTES A ANALISIS DE VIBRACION
(The input data related to vibration analysis)

1 SON DE LA FRECUENCIA PREDOMINANTE	(Are of predominant frequency)
2 SON DE LA PARTE ELECTRICA	(Are of electrical frequency)
3 SON DE LA DIRECCION DE LA AMPLITUD	(Are of the predominant amplitude direction)
4 SON DE LA LOCALIZACION DE LA AMPLITUD PREDOMINANTE	(Are of the predominant amplitude location).
5 SON DE LA RESPUESTA LA AMPLITUD A VARIACIONES DE VELOCIDAD EN SUBIDA	(Are of amplitude changes due to speed increases)
6 SON DE LA RESPUESTA DE LA AMPLITUD A VARIACIONES DE VELOCIDAD EN BAJADA	(Are of amplitude changes due to speed decreases)
7 NO ESTAN DISPONIBLES	(Are not available)

Once the user has answered, the system begins to ask for the data corresponding to each of the options selected. If we assume the user has chosen the options 1, 3, 4, the system asks for the data related to the predominant frequency, the direction of the predominant amplitude, and the location of the predominant amplitude, through the following menus:

LA FRECUENCIA PREDOMINANTE	(The predominant frequency)
1 ES IGUAL A LA FRECUENCIA DE RESONANCIA DEL ROTOR O ESTATOR	(Is the resonant rotor or stator frequency)
2 CAE EN EL INTERVALO DE REMOLINO DE ACEITE (40 - 50%)	(Is on the 40 - 50% frequency range)
3 CAE EN EL INTERVALO DE 50 - 100%	(Is on the 50 - 100% frequency range)
4 ES IGUAL A 1 * FRECUENCIA DE ROTACION	(Is 1X frequency)
5 ES IGUAL A 2 * FRECUENCIA DE ROTACION	(Is 2X frequency)
6 ES IGUAL A MULTIPLOS MUY ALTOS	(Is of higher multiples)
7 ES IGUAL A 1/2 DE LA FRECUENCIA DE ROTACION	(Is 1/2 X frequency)
8 ES IGUAL A 1/4 DE LA FRECUENCIA DE ROTACION	(Is 1/4 X frequency)
9 ES IGUAL A MULTIPLOS MAS BAJOS	(Is of lower multiples)
10 ES IGUAL A FRECUENCIA ASINCRONA	(Is and odd frequency)
11 ES MUY ALTA	(Is a very high frequency)

LA DIRECCION DE LA AMPLITUD PREDOMINANTES ES	(The predominant amplitude direction is)
1 VERTICAL	(Vertical)
2 HORIZONTAL	(Horizontal)
3 AXIAL	(Axial)

LA LOCALIZACION DE LA AMPLITUD PREDOMINANTE SE ENCUENTRA EN	(The predominant amplitude location is at)
1 LA FLECHA	(The shaft)
2 LAS CHUMACERAS	(The bearings)

3 LA CUBIERTA	(The casing)
4 LA CIMENTACION	(The foundation)
5 LA TUBERIA	(The piping)
6 EL COPLE	(The coupling)

When all the data are entered, the system displays a diagnostic report which includes: the certainty probabilitu percentage of the diagnosed failures, and the comments and the recomendations for each one of the conclusions. At user's request, the system is able to show the rules that have been used to reach the conclusions.

Expert Systems for Flue Gas Desulfurization System Operations

K. LUKENS, P. SPERBER, and M. YAMATANI
United Engineers & Constructors
30 South 17th Street P.O. Box 8223
Philadelphia, Pennsylvania 19101, USA

ABSTRACT

This paper presents an Expert System to advise in the operation of Flue Gas Desulfurization (FGD) systems. Specifically, the area of expertise that this Expert System addresses is the water balance of wet FGD systems.

The program is menu driven. It has been designed to have several independent pieces of software interact with one another. This software includes: Expert Ease, a management and graphic package; Nexpert, an expert system shell; Clipper, a dBase compiler; and, a compiled Fortran program.

The program has two modes of operation: (1) Configuration mode where the basic system parameters are initialized (e.g., Tank size, etc.) to determine if basic changes in the system are justified; and, (2) Operation mode where operating parameters can be varied to predict and advise the operator of correct operation of the FGD system. The system in its present form is more fully developed in the operation mode than the configuration mode.

INTRODUCTION

Currently there are approximately seventy flue gas desulfurization systems (FGD) operating at utility installations in the United States. There are three major technical problems associated with operations of these systems. The problems are corrosion, scaling and water balance upsets. Associated with the FGD systems are other contaminated flows which are frequently used as make-up. These include bottom ash water, cooling tower blowdown and treated wastewater. The problems associated with water balance upsets of FGD systems are the subject of this expert system. The major need for an expert system in this area is the lack of a large number of experts for these systems due to their complexity. Also, with current environmental laws and the possibility of future acid rain legislation, managing water balances for the above systems is extremely important.

Figure 1 shows the basic flow streams for a typical conventional lime slurry scrubbing system with integrated waste processing. This is the basis for this version of the expert system. The key areas represented on this diagram are: the FGD absorber island; the thickener system; clarified process liquor tank(s), retention basin (collection of rainfall runoff, maintenance wastes and overflows for recycling); lime preparation system; fly ash conditioning system; and, waste processing system. Liquid inputs to this type of system can include cooling tower blowdown, bottom ash sluice water, and service water. The objective in operating

WATER FLOW DIAGRAM

LIME SLURRY SCRUBBING SYSTEM

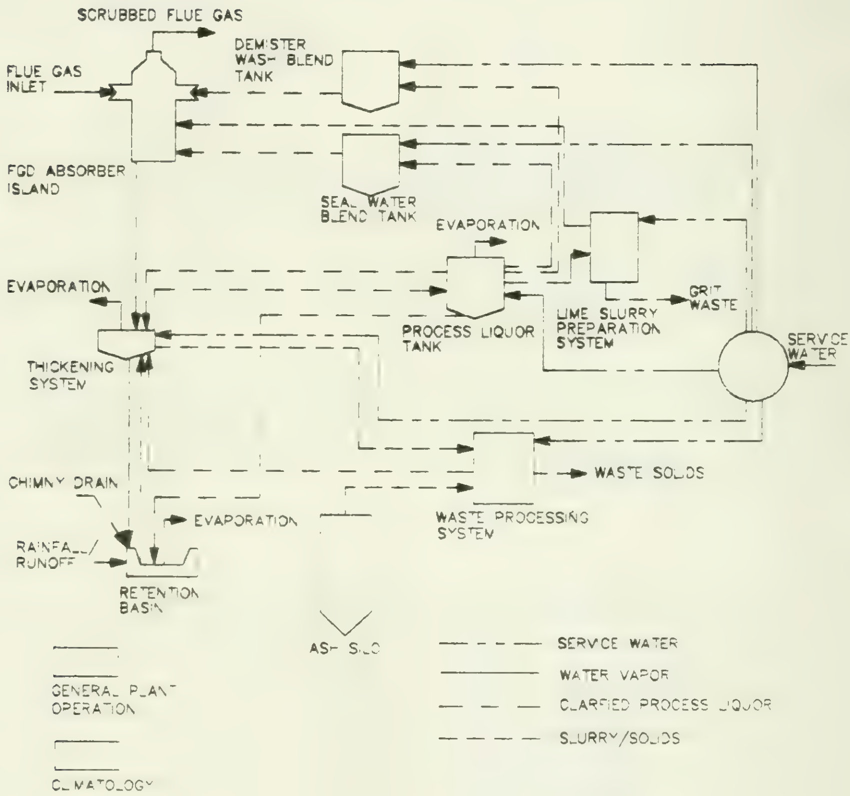


Figure 1

this type of system is to maintain a closed loop water balance under a range of operating conditions. Due to the large number of variables that are affecting the water balance at a given point in time, system operation decisions are based on expertise developed over an extended period of time.

The major problem that can develop in a FGD system when an imbalance occurs is that the zero discharge requirement may be violated if the imbalance is sustained for a sufficient period of time. Such imbalances can also impact operation of the process (e.g., chemistry and scale potential). Frequent water imbalances may also necessitate installation, or expansion, of wastewater treatment equipment.

Currently there are conventional computer tools developed which will calculate a plant water balance based on a set of input conditions. United Engineers has developed a variety of such computer tools for all its recent FGD designs. The next logical step is the development of an expert system to advise both plant operators, plant engineers and FGD system designers of the best approach to either avoid inherent design deficiencies regarding water balance upsets and/or provide guidance for courses of action to recover from imbalances encountered during operation. This expert system is designed to capture the considerable expertise developed over many projects which has a sound engineering basis and to recommend appropriate courses of action to less skilled users. The use of an expert system is a cost effective method of transferring knowledge to plant operations people.

DESCRIPTION OF THE SYSTEM'S FUNCTIONALITY

The system has two modes. These include (1) configuration mode and (2) operation mode. The system provides the diagnostic and advisory support in the operation mode to the operator who monitors the water flow balance of the FGD system. Examples of the types of problems that can be encountered and diagnosed include:

- Absorber island liquid level excursions
- Loss of clarified process liquid tank level control (high and low)
- High retention basin level
- Loss of stable systems chemistry (future incorporation)
- Potential short term loss of overall water balance based upon planned operations
- Absorber demister scale potential from current or planned operations

Examples of operating parameters that can vary are as follows:

- Boiler load
- Demister wash water flow rate and frequency
- Flue gas conditions (temperature, inlet and outlet sulfur dioxide)
- Number of absorber trains in normal operation
- Number of recirculation pumps in normal operation
- Reagent slurry quality (composition)

When an upset occurs, the operator would only change those values which are not the normal condition. The expert system requests the nature of the problem or other information for the planned period of operation through menu selection or direct input. The system then performs a steady state water balance and replies to the user the appropriate course of action based on the available information and the expertise built into the system. The system includes an explanation facility which will advise the user of the reasoning for the recommendation and what changes in the water balance the operator can expect to see and the time frame he should expect to see them. It also shows other possible solutions. The operator can query the expert system for the basis of the selection.

Possible uses of the expert system by plant operators or engineers would be as a trend analyzer to predict what the condition would be over extended holiday periods, unusual rainfall periods or the decision on what will be the effect if certain equipment is removed from service for maintenance while the system continues to run. The use of the system as a prelude to possible system modifications would be to predict the benefit or the negative effect of these modifications.

In the configuration mode, the system allows input of the "plant specific conditions". The quantity of data is sufficient to address most, but not all, specific plant configurations. Examples of plant specific configuration information are the following:

- Number of absorber trains
- Number of recirculation pumps per absorber
- Number of agitators per absorber
- Waste processing filtration capacity
- Type of demister water (cooling tower blowdown, service water, etc.)

The system in the configuration mode when fully developed will allow the plant engineer to perform planning analysis. There are two function to this: (1) as a trend analyzer, and (2) as a prelude to possible system modifications.

DESCRIPTION OF PROGRAM

The system is menu driven. When the system is started, the user can select the worksheet, which for this case is the diagram shown in Figure 1. A mouse input device is used to activate pop-up menus by locating the cursor on the component when the parameters are to be adjusted. The components and environmental operating conditions that can be changed include the following:

1. FGD Absorber Island
2. Demister Wash Blend Tank
3. Seal Water Blend Tank
4. Lime Slurry/Preparation System
5. Process Liquour Tank
6. Thickener System

7. Retention Basin

8. Service Water

9. Ash Silo

10. Waste Processing System

11. General Plant Operation

12. Climatology

A typical example of the information that can be changed is the General Plant Operating Conditions. The items that can be varied include plant load, fuel composition, excess air, heat rate, the season of the year, and flue gas conditions (temperature and pressure).

After the parameters have been selected, a Fortran program is then run to calculate the overall water balance of the FGD system. After the calculations have been made, the Expert system program will be activated to advise on the status of the system and to provide a recommended course of action concerning the water balance, if necessary. Where there are several courses of action, the expert system presents them in order of preference. The expert system includes an explanation facility to state the reasoning being used. At this point the user can accept the advice and recalculate the water balance based on the expert system recommendations or recalculate using the Fortran program using other values.

The output from a session using this program includes both an output report and/or graphic representation of the water balance as a function of time. This information can be utilized by the operator to make choices concerning which sources of make-up to the system, where there is a control options over a given time period. This will preclude unnecessary discharges from the system.

EQUIPMENT REQUIRED FOR THE PROGRAM

The program can be run on conventional personal computer hardware consisting of an IBM 286 or clone with a minimum of 2 MB RAM. The processor speed should be 16 or 20 MHz. The program will also run on an IBM 386 machine. The size of the hard drive should be a minimum of 40 MB since there will normally be an extensive amount of information associated with this type of system.

CONCLUSION

The paper describes an expert system developed to address the need to advise operators and designers of FGD systems concerning the important issue of water balance of the systems. This expert system can act as a useful tool to advise inexperienced operators and ultimately designers in this domain of knowledge. It can also be the shell to expand into FGD operation chemistry control.

NUCLEAR PERFORMANCE APPLICATIONS

Robust Handling of Dynamics and Multiple Failures in a Diagnostic Event Analyzer

F. ERIC FINCH and MARK A. KRAMER
Laboratory for Intelligent Systems in Process Engineering
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA

ABSTRACT

MIDAS diagnoses malfunctions in continuous chemical and refinery processes using a plant-independent strategy based on qualitative and quantitative process models. MIDAS specifically addresses problems not treated in past systems, including: process dynamics with control system responses, multiple faults and induced failures, and out-of-order and false alarms. This paper discusses both the structure of the process models and the diagnostic reasoning strategies employed by MIDAS.

INTRODUCTION

Fault diagnosis involves detection of abnormal process conditions and identification of their root causes. Two classes of root causes can be considered: equipment degradation and failure including heat exchanger fouling, sensor bias and failure, pipe blockages, catalyst deactivation, and leaks, and external disturbances such as feedstock or utility variations. We will refer to both classes as process malfunctions.

Diagnosis of process malfunctions can be a difficult task for the process operator, making operator aids of interest. Such aids can take many forms, such as increased instrumentation or improved human-machine interfaces. However, in this paper we focus on operator aids based on advanced information processing technology involving artificial intelligence (AI).

AI techniques are of particular interest in this domain, because of difficulties in applying conventional numerical methods. These difficulties arise primarily from practical limitations in preparing accurate mathematical models and excessive computational requirements. AI overcomes these limitations using either "deep knowledge" techniques in which detailed mathematical models are replaced by qualitative or approximate models and rigorous mathematical solution techniques with logical and heuristic methods, or "shallow knowledge" techniques which dispose of explicit models entirely, in favor of diagnostic rules-of-thumb.

Still, for dynamic systems, diagnosis remains a difficult problem to address even using AI techniques. Some of the outstanding problems in dynamic systems diagnosis by AI techniques include the following:

1) Utilization of Evolving Information. Temporal features in the dynamic evolution of a malfunction contain important diagnostic information (1, 2). However, most methods are designed to interpret a "snapshot" of plant states at a single time point. Since process dynamics can exhibit non-monotonic behaviors including compensatory (e.g. normal -> high -> normal) and inverse (e.g. normal -> high -> low) responses, a snapshot is potentially misleading. A diagnostic system utilizing dynamic information must incorporate knowledge of the full range of dynamic behaviors produced by the plant, including the effects of control systems and other feedback mechanisms.

2) Robustness to Symptom Variation. When disturbances caused by a malfunction propagate through the plant, they rarely follow an exactly predictable pattern. The temporal sequence of discrete state indicators such as alarms can be influenced by the malfunction extent versus time profile, the location of the malfunction, sensor noise, and the decision thresholds associated with individual alarms (1). Like operators, a diagnostic system that utilizes discrete states or events must be capable of coping with such realities as missing or out-of-order alarms.

3) Diagnosis of Multiple Malfunctions. Multiple malfunctions can be simultaneous independent malfunctions, or induced failures (malfunctions caused by other malfunctions). Although the general multiple malfunction diagnosis problem remains intractable, a reasonable goal is to deal with induced failures of sensors (which account for most induced failures), and simultaneous, independent, and non-overlapping failures. The latter assures that unrelated alarms (including false alarms) occurring during the diagnosis of another malfunction do not destroy the ability to diagnose accurately.

In this paper, we describe the Model-Integrated Diagnostic Analysis System (MIDAS), which has been specifically designed to overcome these problems. MIDAS is a model-based, deep knowledge system that performs diagnostic analysis independent of operator input (i.e. runs as an operator associate, not a consultant) using available on-line measurements. In the following sections, we describe the basic strategy for diagnosis implemented in MIDAS, the classes of objects defined, and the details of the inference cycle. A discussion of how these choices facilitate the treatment of dynamics, out-of-order alarms, and multiple faults then follows.

METHODOLOGY FOR DYNAMIC SYSTEMS DIAGNOSIS

The diagnostic inference methodology implemented in MIDAS is modeled on the deductive reasoning process of an operator performing diagnosis in a dynamic environment. This model is idealized to the extent that it originates from no specific operator. Instead, the model derives from "putting ourselves in the operator's shoes," and determining the optimal actions and conclusions in certain well-defined hypothetical situations. This analysis gives MIDAS its methodology (the sequence of actions and steps for performing the diagnosis) and its ontology (the definition of the objects, concepts, and relationships to be represented).

Imagine the process operator in a central control room monitoring a set of alarm annunciators. These alarms can be associated with any observable process condition that is deemed diagnostically useful, and do not necessarily correspond to the alarms of the process control system. Possible examples of alarms include qualitative states (TEMPERATURE HIGH), trends (PRESSURE INCREASING), equation residuals (MASS BALANCE VIOLATED), inequalities (SIGNAL OUT OF RANGE),

or noise characteristics (SIGNAL FLAT). The alarms incorporate a definition of the nominal steady state or the trajectory defining normal operation, and indicate significant deviations from the normal condition.

An event is defined as any change in alarm status (an alarm turning on or off). If we ascribe to the operator the ability to reason rapidly, events can be analyzed one at a time in their order of occurrence. The operator will respond to each event as it occurs, and conduct the diagnosis from the sequence of events.

Assume that the operator possesses knowledge of the likely or important potential malfunctions, and a mental model of plant behavior consisting of causal relationships among malfunctions and events. When a new event occurs, this knowledge is applied with the goal of explaining the cause of the new event in terms of new or existing malfunction hypotheses. If the operator successfully performs this analysis, the result will be:

- A hypothesis concerning the number of independent malfunctions,
- A list of hypothesized root cause candidates for each of the malfunctions,
  and
- Hypothesized causal explanations for each past event.

Upon detection of subsequent events, the operator can either re-start the reasoning from scratch, discarding all previous hypotheses, or utilize the existing hypotheses to help interpret new events. It is clearly more efficient to do the latter, that is, analyze new events in the context of existing hypotheses. New events can affirm or deny existing hypotheses, or can compel the creation of new hypotheses.

Therefore, for each new event, the operator determines the cause(s) of the event, and updates existing malfunction hypotheses. Although different parts of the model knowledge are exercised, the goals and procedures applied to each new event are the same.

There are several noteworthy characteristics in this model of diagnostic performance:

1) It is always focused on interesting events. In the idealized model, the task of the operator is always to explain the most recent event. Measurements are routinely scanned for alarm situations, but in-depth diagnostic analysis is reserved for events.

2) It is semi-continuous, rather than batch. Unlike the standard consultation mode of expert systems, in this model, hypotheses evolve as new information is revealed, rather than being re-created from scratch with each successive snapshot of the plant state. Therefore, it is efficient and well-suited for the dynamic environment.

3) It is sensitive to the sequence of events. Since new events are analyzed in the context of existing hypotheses, the conclusions are a "path function," sensitive to the sequence of events, rather than a "state function". Diagnostic information contained in the temporal order of events is not lost.

4) No "expert" rules are required. We have assumed that the operator uses only a qualitative model of the process (deep knowledge) to perform the diagnosis. This is an oversimplification, but there many situations where shallow knowledge is either unavailable, unverifiable, or

incomplete (1). The qualitative model, on the other hand, can be derived systematically from the flowsheet of the plant (3). For more discussion of deep versus shallow knowledge approaches, see Fink and Lusth (4).

5) The methodology is plant-independent. Provided a causal model of specified format can be developed and normal process operations can be characterized, the methodology will apply regardless of the actual nature of the process.

MODELING CONSIDERATIONS

One guiding doctrine of AI is to utilize an explicit representation of the objects in a problem domain. This includes not only physical objects such as tanks, pumps, and pipes, but conceptual objects as well. Examples of conceptual objects in other domains are design goals, fault trees, control objectives and constraints, success paths (nuclear plant operations), and the "pinch" in heat exchanger networks. In our model of the diagnostic process, the relevant conceptual objects include hypotheses, events, and causal links. In MIDAS, physical and conceptual objects are represented using frames (5). Frames permit the explicit, formalized definition of an object and an associated set of properties. The formalized declaration of the objects and relationships in a domain is the ontology of that domain. In MIDAS, the ontology consists of three major categories of objects: monitors, process model, and hypothesis model. Figure 1 diagrams the interrelationships of these elements within MIDAS.

Monitors

Monitors act as the "alarms" within MIDAS. There are two types of MIDAS monitors: sensor monitors which collect and analyze data from on-line process sensors and constraint monitors which collect and analyze constraint residuals computed from sensor data. For every process sensor or constraint equation there is a monitor that exists as an object in the MIDAS knowledge base.

The purpose of the monitors is to translate numerical measurements and constraint residuals into events. To detect events from process data, a monitor must know both the nominal value or trajectory for the measured variable or constraint, and the normal variability produced by measurement noise and ordinary transients. These are used as a basis for comparison in analysis of incoming data. Sensor monitors can also produce events related to gross sensor failures, using sensor range and noise limit checks. Violation of these checks result in a type of event that in the inference process is linked directly with sensor failure[1].

Constraint monitors are the means by which MIDAS exploits the information contained in equality and inequality constraints. The value of quantitative constraints in diagnosis is well established (6, 7, 8). In MIDAS, algebraic combinations of measurements are treated as "virtual sensors". For example, the residual of a mass or heat balance has a nominal value and operating range and can be monitored like a sensor. Constraint monitors produce events in exactly the manner of a sensor monitor when the residual in question undergoes a significant deviation. Monitors apply statistical criteria to detect events.

---

[1] If range and noise tests are passed, the sensors are not automatically validated; bias and in-range failure are considered in subsequent reasoning.

Two types of analysis are performed: appraisal and prediction. Appraisal examines a vector of measurement data and applies tests similar to statistical quality control tests (9) to identify abnormal trends or shifts in values. These tests indicate an event if one data point is outside a control limit, or two consecutive data points are outside a warning limit. Sensor failure events will be indicated if n consecutive data points are identical, if any data point is outside the allowable range, or if a significant change in noise variance is detected.

Prediction entails making a forecast of future measurement values by extrapolating from a fit of past data. Many features of the prediction are user modifiable such as the type of fitting function and the forecast time horizon. Forecast values are subjected to tests similar to those used in appraisal to determine if an event can be expected to occur in the future. Prediction is used primarily as a robustness feature in conjunction with interrogation which is discussed subsequently.

In our implementation, monitor tests indicate current events with approximately 95% confidence and expected events with approximately 50% confidence. Test criteria can be adjusted by the user. Alternate techniques could also be applied to detect events (10).

MIDAS monitors are implemented as frames, with demons (active values) invoking attached procedural code whenever new data is received. When a monitor determines that a new event has occurred, the monitor activates the event interpreter, discussed below.

Process Model (PM)

The causal model specific to the current process is contained in the PM. The PM consists of a structured description of potential root causes and events, and the causal relationships among them. The model is constructed prior to use and undergoes no structural modifications during diagnosis. The PM represents the user's main input to MIDAS.

The starting point for derivation of the PM is a process signed directed graph (SDG) assembled from component subgraphs. A qualitative matrix analysis is applied to determine the non-monotonic responses implicit in the SDG, resulting in an extended sign directed graph (ESDG) (11). Another procedure converts the ESDG into a PM by eliminating unmeasured variables and coalescing causal pathways (3). The effort necessary to construct the PM using this technique is significantly less than that required to produce the PM manually. The systematic procedure is also less subject to errors than manual creation of the PM.

The PM consists of the following object types:

1) Measured Variable (MV) and Measured Constraint (MC). These objects contain information related to observed variables and constraints, including:

   - Current status,
   - Current state and history of past states,
   - Current trend and history of past trends.

   In this context, status is the functional state of the sensor(s) observing the MV or MC. Status is OK if the sensor reading is judged reliable or FAILED if the monitor has detected a gross sensor failure. The state of a variable or constraint can have values HIGH, NORMAL, or LOW, and the

trend can have values INCREASING, STEADY, or DECREASING.

2) Potential Event (PE). PE objects store information relating to possible events. Events are changes in status, state, or trend of the monitored object. For example, a variable with high and low alarms will have four associated events: HIGH-ALARM-ON, HIGH-ALARM-OFF, LOW-ALARM-ON, and LOW-ALARM-OFF. It is necessary to specify for each instance of a PE the following information:

   - Name of the associated variable or constraint,
   - Status, state, and trend prior to the event,
   - Status, state, and trend subsequent to the event,
   - Exclusive events.

   Exclusive events are other PEs which cannot be present at the same time as the PE in question. For example, HIGH-ALARM-ON and HIGH-ALARM-OFF are exclusive events.

3) Potential Root Cause (PRC). The PRC objects enumerate the potential malfunctions of the plant. Important attributes of potential root causes include:

   - Tests that can be applied to confirm or deny the presence of the malfunction,
   - Prior probability of the malfunction.

4) Local Cause Link (LCL). This type of object represents a causal relationship between a PRC and a PE. These links define the primary symptoms of each malfunction, that is, the PEs expected first when the malfunction is present.

5) Precursor/Successor Link (PSL). A PSL represents a directional causal link between two PEs. Attributes of PSL objects include:

   - Status, state, and trend of both PE objects linked by the PSL,
   - Diagnostic conditions that apply if the link is invoked during the reasoning process,
   - Conditions under which the link is inactive.

Hypothesis Model (HM)

The HM is constructed on-line by the MIDAS reasoning process. The HM is the active part of MIDAS' memory, keeping a record of observed events and current diagnostic hypotheses. The HM consists of objects that are created during the monitoring of the plant.

The objects represented in the HM include malfunctions and events, similar to the PM. However, the HM contains not potential events and root causes, but rather, events and root causes that have been observed or hypothesized. The clarity of MIDAS is improved by separation of potential and actual objects. An additional benefit relates to truth maintenance. Truth maintenance is the task of managing assumptions, so that when an assumption is retracted, the conclusions derived from the assumption will also be retracted (12). In real-time expert systems, truth maintenance takes on the additional element of time dependency. Conclusions may need to be retracted after the expiration of a "validity interval," based on the time scales of changes in underlying variables. Complicated agenda mechanisms are required to schedule data acquisition in coordination with validity intervals. MIDAS, however, avoids the problems

associated with dynamic truth maintenance by representing facts in an event-oriented format. Once deduced, a MIDAS event such as "REACTOR TEMPERATURE SENSOR HIGH AT 13:00 HOURS" remains true forever. The event is a transition that occurred at a specific past time point and cannot be undone. If reactor temperature subsequently returns to normal, another event -- "REACTOR TEMPERATURE SENSOR NORMAL AT 13:10 HOURS" -- is deduced. Both events can exist without conflict because they occurred at different time points.

After a malfunction episode, MIDAS retains a record of the disturbance in the form of a sequence of observed events. This record can later be reviewed to follow the complete course of the malfunction, and track the reasoning performed by MIDAS.

The following objects comprise the HM:

1) Recorded Event (RE). A RE represents an observation of a potential event. A new RE object is created whenever an event is detected by the monitors. Attributes of a RE include:

   - Time of observation,
   - Confidence of observation,
   - Corresponding potential event object,
   - Classification as either a primary or secondary symptom.

2) Hypothesized Root Cause (HRC). A hypothesized root cause is a potential root cause hypothesized to be present in the process. Hypothesized root cause attributes are:

   - Corresponding potential root cause object,
   - Recorded and expected events that support and oppose the HRC,
   - Relative likelihood of the HRC with respect to other HRCs.

3) Inferred Malfunction (IM). It may not be possible or desirable for the event interpreter to link all REs in the HM into a single connected structure, either because no causal pathway exists between the events or because the pathway contains multiple intervening unobserved events. As a result, the HM typically consists of multiple clusters of connected REs. We refer to these clusters as inferred malfunctions. Figure 2 illustrates IM clusters. Each IM object represents a group of causally related observations, with a presumed common cause. Attributes of IM objects include:

   - A list of REs associated with the cluster,
   - A list of HRCs associated with the cluster.
   - A classification of the malfunction type as PERSISTENT, PERIODIC, ONGOING-TRANSIENT, COMPLETED-TRANSIENT, or CORRECTED,
   - A list of possible source events.

   Here, source events are REs, that through causal links can explain all other REs in the IM cluster.

INFERENCE IN MIDAS

The event interpreter is a set of procedural algorithms designed to construct hypotheses on the causes of events. The interpreter draws on knowledge of potential malfunctions and their effects contained in the PM and knowledge of past observed events and the current malfunction hypotheses contained in the HM. Figure 3 is a condensed flowchart of the event interpreter. The interpreter is

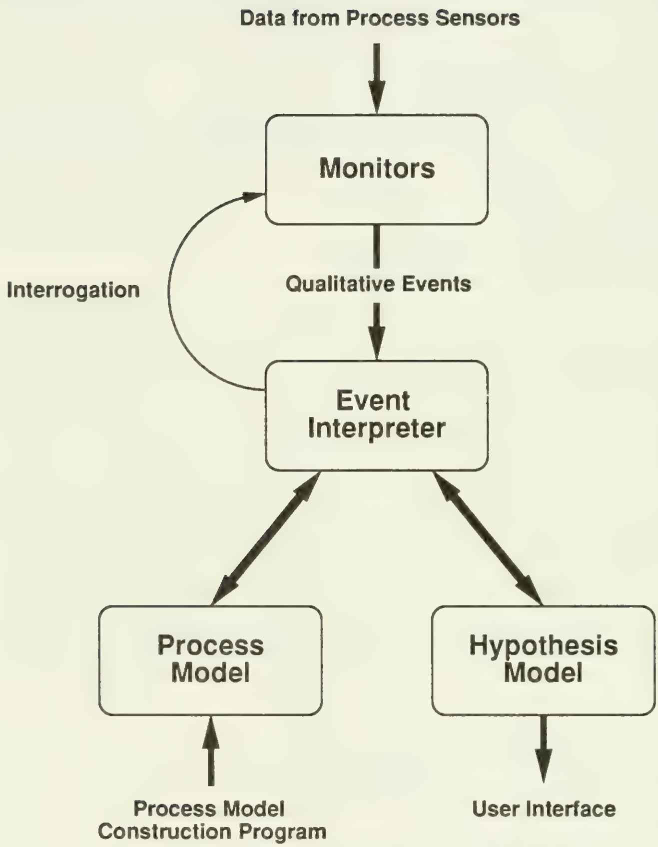


Figure 1. Architecture of MIDAS

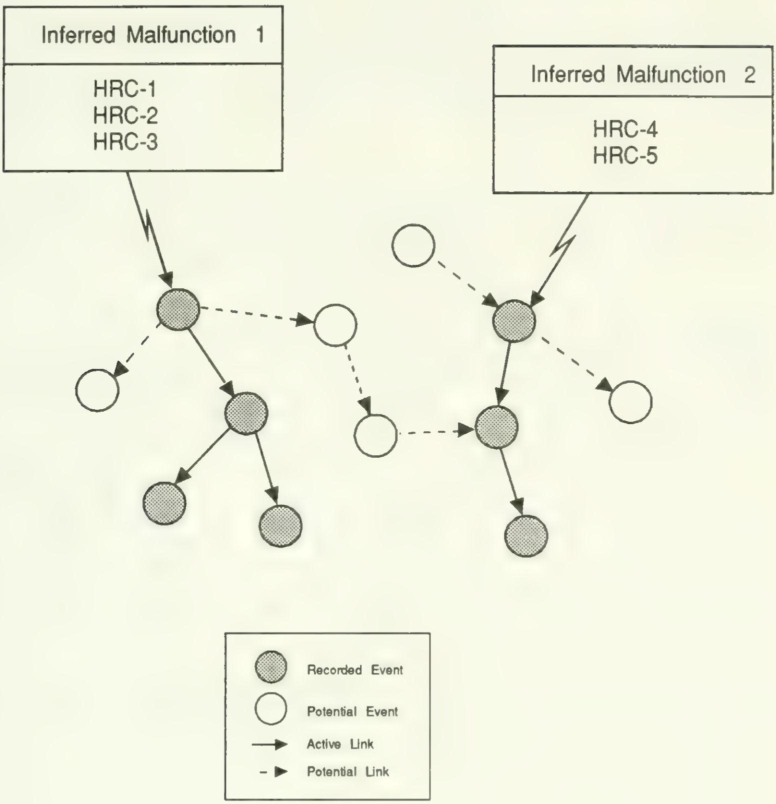


Figure 2. Relationship of Inferred Malfunctions, Recorded Events, and Hypothesized Root Causes (HRCs)

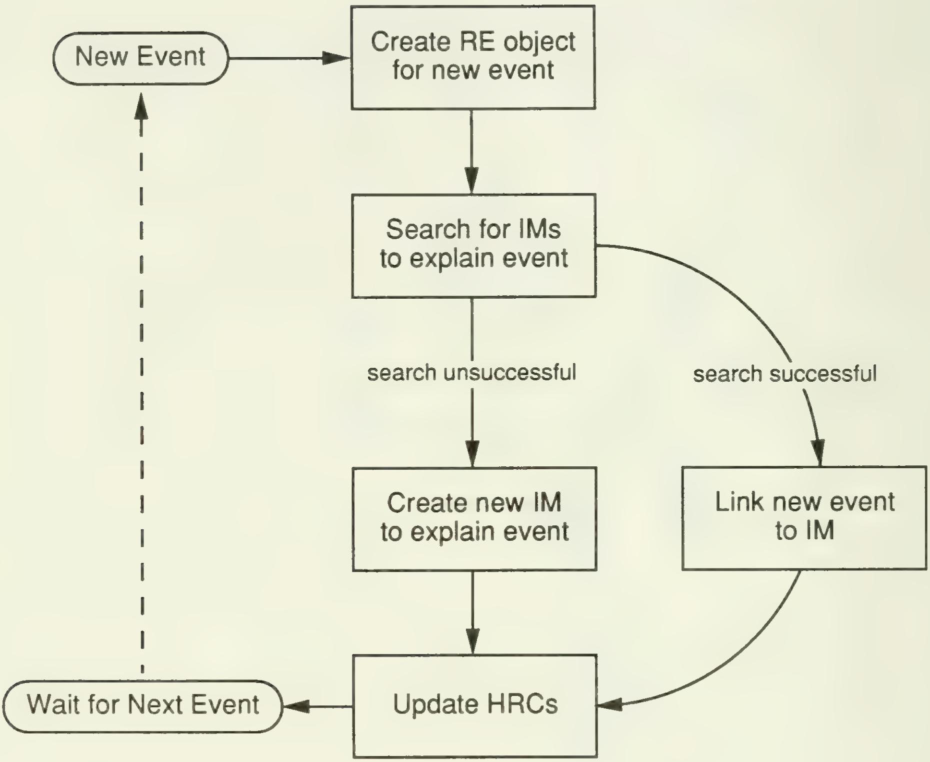


Figure 3. Flowsheet of Inference Cycle of MIDAS

triggered by the detection of a new event in the monitors. The first action following detection is creation (instantiation) of a new recorded event object, and assignment of values to the slots for the type and time of occurrence of the event. This is analogous to the operator registering that there has been a new alarm.

The next task is to determine whether the event is a symptom of a known malfunction, or indicative of a new malfunction. This determination is made by searching for precursor/successor links relating the new event to existing REs, and by searching for local cause links between the new event and active HRCs. Missing and out-of-order events are accounted for at this stage using procedures discussed later. If a link is found, the event is assumed to be the consequence of an existing malfunction[2], and the new RE joins an existing inferred malfunction cluster. If no causal connection is found, it is assumed that a new malfunction is present, and the event becomes the sole RE of a new IM cluster.

The next step is to determine if new HRCs should be created, by determining whether the new event is a source event (this occurs only for new clusters and for certain situations of reversed event sequences). If the new event is a source event, the root causes which list the new event as a primary symptom are instantiated as new HRCs. If not, no new HRCs are added.

Finally, the likelihood ratings of the HRCs in the affected cluster are re-evaluated. This involves updating the supporting evidence (SE) and opposing evidence (OE) of each HRC, and then recalculating the likelihood of each root cause hypothesis. The SE and OE are the recorded and expected events in the cluster that support and oppose the hypothesis, respectively, based on the existence, or lack of, a causal pathway between the root cause and the event. The following formula is used for calculating relative likelihoods:

$$R_i = P_i \cdot \prod_{j}^{OE} \alpha_j \cdot \prod_{j}^{SE} (1 - \alpha_j) \qquad (1)$$

where R_i is the likelihood rating of the i-th hypothesis ($0 \leq R_i \leq 1$), P_i is the prior probability of the root cause, and α_j is the probability of false detection of the event.

Ranking the hypotheses completes one "cycle" of the event interpreter. As much information as can be deduced from the available data has been derived, and the event interpreter can now wait for the next event.

DISCUSSION

What follows is a brief discussion of several problems that have been addressed in MIDAS.

---

[2] The possibility that two or more malfunctions can lead to symptoms attributable to one malfunction is not entertained in MIDAS. MIDAS therefore produces a minimal set of malfunctions spanning the set of recorded events.

Dynamics

With the proper construction of the PM, MIDAS will correctly interpret control system responses and other non-monotonic behaviors. This is accomplished by creating precursor/successor links between potential events of the same variable as illustrated in example 1.

Example 1. Assume that reactor temperature (T_r) is a controlled variable subject to disturbance by a variety of malfunctions. If a malfunction occurs that tends to increase T_r, one possible observed behavior involves a temporary increase in T_r, with two recorded events: "T_r HIGH AT 0:00 HOURS" and "T_r NORMAL AT 0:05 HOURS".

To indicate the latter event is an expected control system response, a precursor/successor link exists in the PM between the potential events T_r HIGH and T_r NORMAL. This link is constructed by the PM construction algorithm previously mentioned. When T_r NORMAL is observed, MIDAS attributes the behavior to the previous event involving T_r HIGH, retaining the HRCs associated with T_r HIGH. If such a link had not been present in the PM, MIDAS would interpret the behavior as arising from either a transient malfunction or a false alarm.

Out-of-Order Events

Alarm systems in general cannot simultaneously achieve sensitivity and guarantee detection of events in causal order (see Appendix A). As a result, if the primary goal of an alarm system is sensitivity to deviations from normality, the possibility of out-of-order alarms must be accepted. It is therefore necessary for a diagnostic system to build in robustness to variations in event sequences.

MIDAS approaches the problem of out-of-order events through various features included in the inference algorithm. These features can be divided into two categories: anticipation and correction.

When a new event can be linked to previously observed events only through paths traversing one or more unobserved events, the interpreter will try to determine if the intervening undetected events are "out-of-order" and can be anticipated in the near future. The interpreter interrogates the monitors responsible for detecting the intervening events and asks for a prediction. If the monitors indicate that based on present trends the missing events are likely to occur within a pre-defined time horizon, the interpreter completes the link, behaving as if the intervening events had already been detected. Otherwise, the link to previous events is not established and a new IM will be created for the new event.

Since prediction is probabilistic in nature, it is sometimes necessary to correct erroneous predictions. For example, if interrogation reveals an event is not expected, and the event subsequently occurs, the cluster created as a result of the erroneous prediction is joined to the original cluster and the HRCs are revised appropriately. MIDAS cannot, however, correct itself when a monitor gives an erroneous prediction of a future event. This limitation stems from the fact that MIDAS lacks a time-based agenda, such as that in the

G2[3] system, which would be needed to schedule future checks on the validity of the expected event assumption. If MIDAS had such a capability, and the expected event did not occur as predicted, a procedure reversing the assumption would be carried out.

Multiple Malfunctions

The capability to handle multiple faults in MIDAS derives directly from the ontology of the hypothesis model. Separate event clusters are maintained under different IMs, allowing the system to work simultaneously on more than one cluster of events. If the IM structure were eliminated (say if hypotheses were contained in an unstructured list), then it would not be possible to represent multiple malfunctions.

Multiple malfunctions will be postulated if the event interpreter fails to link a new event to existing inferred malfunctions, as described earlier. MIDAS will successfully diagnose multiple malfunctions if the malfunctions are sufficiently distant from each other that their symptoms do not overlap or cancel out. Here, distance refers to the separation in the PM.

MIDAS will also successfully diagnose multiple malfunctions if one of the malfunctions is a gross sensor failure, independent of the degree of separation. This is because gross sensor failure is diagnosed by checks in the monitors and much of the analysis of the event interpreter is bypassed. This property is useful in handling induced sensor failures.

In other cases where symptoms from two malfunctions overlap, MIDAS may assume a common cause with unpredictable results.

IMPLEMENTATION

MIDAS has been implemented in GoldWorks[4] on a 386-class PC and makes use of its frame structures and functions to represent and process knowledge. Inferencing is accomplished using a combination of message passing, active slot values, and custom LISP routines. The expert system inference engine included in GoldWorks is not used, and no knowledge is stored in IF-THEN rules.

A data acquisition interface has been developed so MIDAS can access DBASE III[5] records. For presentation of results, a menu-driven interface allows inspection of information contained in PM, HM, and monitor objects. From the interface, the user can also modify inference parameters such as event detection probabilities and search depths.

CONCLUSIONS

This paper has summarized the basic structure and features of the MIDAS diagnosis system and described how MIDAS handles the complications imposed by process dynamics, out-of-order events, and multiple faults. Success in handling these problems derives from the combination of a new ontology and a methodology adapted to the problems of diagnosis in the dynamic process environment.

---

[3] Trademark, Gensym Corporation, Cambridge, MA.
[4] Trademark, Gold Hill Computers, Cambridge, MA.
[5] Trademark, Ashton-Tate, Torrance, CA.

ABBREVIATIONS

AI	--	Artificial Intelligence
ESDG	--	Extended Signed Directed Graph
HM	--	Hypothesis Model
HRC	--	Hypothesized Root Cause
IM	--	Inferred Malfunction
LCL	--	Local Cause Link
MC	--	Measured Constraint
MIDAS	--	Model-Integrated Diagnostic Analysis System
MV	--	Measured Variable
OE	--	Opposing Evidence
PE	--	Potential Event
PM	--	Process Model
PRC	--	Potential Root Cause
PSL	--	Precursor/Successor Link
RE	--	Recorded Event
SDG	--	Signed Directed Graph
SE	--	Supporting Evidence

ACKNOWLEDGMENTS

The research was supported by National Science Foundation Grant CBT-8814226. The authors would also like to thank Olayiwola O. Oyeleye for his major contributions to MIDAS in the area of process modeling.

REFERENCES

1. M.A. Kramer. "Expert Systems for Process Fault Diagnosis: A General Framework." Proc. First Intl. Conf. on the Foundations of Computer-Aided Process Operations, Park City, UT, G.V. Reklaitis and H.D. Spriggs, eds., Amsterdam: Elsevier, 1987, p. 557.

2. T.M. Washio, M. Kitamura and K. Sugiyama. "Development of Failure Diagnosis Method Based on Transient Information of Nuclear Power Plant." J. Nucl. Sci. Tech., Vol. 24, 1987, p. 452.

3. O.O. Oyeleye. Sc.D. Thesis, Cambridge, MA: Massachusetts Institute of Technology, 1989.

4. P.K. Fink and J.C. Lusth. "Expert Systems and Diagnostic Expertise in the Mechanical and Electrical Domains." IEEE Trans. Sys. Man Cyber., Vol. SMC-17, 1987, p. 340.

5. R. Fikes and T. Kehler. "The Role of Frame-Based Representation in Reasoning." Commun. ACM, Vol. 28, 1985, p. 904.

6. J. Gertler and D. Singer. "Augmented Models for Statistical Fault Isolation in Complex Dynamic Systems." Proc. Amer. Control Conf., Boston, MA, 1985, p. 317.

7. P.S. Dhurjati, D.E. Lamb and D.L. Chester. "Experience in the Development of an Expert System for Fault Diagnosis in a Commercial Scale Chemical Process." Proc. First Intl. Conf. on the Foundations of Computer-Aided Process Operations, Park City, UT, G.V. Reklaitis and H.D. Spriggs, eds., Amsterdam: Elsevier, 1987, p. 589.

8. J. Rosenberg, R.S.H. Mah and C. Iordache. "Evaluation of Schemes for Detecting and Identifying Gross Errors in Process Data." Ind. Eng. Chem. Res., Vol. 26, 1987, p. 555.

9. W.A. Shewhart. Economic Control of Quality in Manufactured Product. New York: Van Nostrand, 1931.

10. K. Liu and J. Gertler. "A Supervisory (Expert) Adaptive Control Scheme." Proc. IFAC 10th World Cong. Auto. Control, Munich, FRG, 1987, p. 37.

11. O.O. Oyeleye and M.A. Kramer. "Qualitative Simulation of Chemical Process Systems: Steady-State Analysis." AIChE J., Vol. 34, 1988, p. 1441.

12. J. DeKleer. "An Assumption-Based TMS." Artificial Intelligence, Vol. 28, No. 2, 1986, p. 127.

APPENDIX A: NOTE ON ALARM SYSTEM DESIGN

It can be shown with a simple example that, in general, an alarm system cannot simultaneously satisfy sensitivity targets and guarantee in-order events, if the extent versus time profile of the malfunction is unknown at the design stage. We presume the purpose of the alarm system in this context is to detect deviations from normality, not to alert the operator to unsafe states.

Consider two measured variables x and y related by a first order transfer function, $G(s) = y(s)/x(s) = K_p/(\tau s + 1)$. Assume the sensor noise on x is characterized by a standard deviation σ_x, and y by σ_y. Let the alarm threshold on x be x*, and similarly y* for y. For each alarm, the following constraints are imposed on the thresholds:

$$U_x > x^*/\sigma_x > L_x \quad \text{(A1)}$$

$$U_y > y^*/\sigma_y > L_y \quad \text{(A2)}$$

The lower bound L corresponds to the maximum tolerable false alarm (type I error) rate. Conversely, U limits the rate of type II errors, the failure of the alarm system to detect an event when an event is present. U is related to the minimum acceptable sensitivity of the alarm system.

For a large, fast disturbance forcing a rapid change in x (such that dlnx/dt >> $1/\tau$), the alarm on x will ring before y, since y cannot respond on such a rapid time scale. If, however, a malfunction were to cause a slow change in x, the steady state relation between y and x, $y/x = K_p$, prevails. If the alarm on x is required to ring before the alarm on y, then at some time y* > y and x > x*. Therefore, the following constraint guarantees in-order operation of the alarms:

$$y^* > K_p x^* \quad \text{(A3)}$$

A feasible alarm setting for y exists if and only if (A1) - (A3) are simultaneously satisfied. Specifically, for y* to exist,

$$U_y \sigma_y > K_p L_x \sigma_x \quad \text{(A4)}$$

Since the sensor variances and the process gain are process-specific, satisfaction of (A4) cannot be guaranteed.

Process Fault Diagnosis Using Knowledge-based Systems

A. L. SUDDUTH
Duke Power Company
P.O. Box 33189
Charlotte, North Carolina 28242, USA

Abstract

Advancing technology in process plants has led to increased need for computer based process diagnostic systems to assist the operator. One approach to this problem is to use an embedded knowledge based system to interpret measurement signals. Knowledge based systems using only symptom based rules are inadequate for real time diagnosis of dynamic systems; therefore a model based approach is necessary. Though several forms of model based reasoning have been proposed, the use of qualitative causal models incorporating first principles knowledge of process behavior, structure, and function appear to have the most promise as a robust modeling methodology. The structure of a diagnostic system is described which uses model based reasoning and conventional numerical methods to perform process diagnosis. This system is being applied to emergency diesel generator systems in nuclear stations.

Introduction

The advance of technology in the process industries has increased the demands on the operators of processes in power stations, chemical plants, and refineries. This increase is due to the complexity of new processes, the economic and environmental consequences of process malfunctions, the initial low level of expertise for operators, and the need to increase reliability and quality of process operation in order to reduce the cost of the products of processes. Typical characteristics of process complexity include control rooms with thousands of annunciator alarms, CRT based operator control stations with hundreds of graphical images, geographically large process plants with fewer operators available for hands-on monitoring of component performance, and increased instances of the use of complex control schemes which provide much less conceptual support for the operator in determining what the controls are doing and whether the process is operating normally.

In order that a process be properly designed, it must be possible to prevent catastrophic failures due to process faults. It is the responsibility of the process monitoring and control systems (human or machine based) to provide appropriate response. The ability of the operator or of the supervisory control system to provide that response depends on rapid recognition of the nature of the situation. The actions required of the combination of human and machine in response to faults can be characterized as some combination of these five tasks (Isermann, 1983):

- Process fault detection - the act of determining that a fault condition exists; that process states are no longer within allowable limits.

- Process fault diagnosis - the act of locating a fault and identifying its cause.

- Process fault evaluation - the act of determining the proper action to take in the event of the fault. This implies the presence of some form of fault classification system which maps identified faults into a set of responses appropriate to the type and level of severity of the fault.

- Process fault mitigation - once a fault is detected and diagnosed, and a proper response to the fault determined, action must be taken by the operator or the supervisory system.

- Process fault management success assurance - The operator or the supervisory system must then followup to ensure that the action taken has been effective in response to the fault.

It is common in many situations for operator action to be performed prior to a complete diagnosis of the nature of the fault because of the urgency of operator intervention. The action in this case consists of the maintenance of certain critical functions associated with protection of the process and the environment. The idea of fault response based on symptoms rather than diagnosis is the basis for much of the recent work in preventing nuclear power station catastrophes (Betancourt, 1984, Gaudio and Jamieson, 1987). Such symptom based response is effective only when the complete range of possible faulted behaviors has been discerned explicitly, a costly and difficult to verify task. In addition, an ultimate diagnosis is required prior to resuming normal operation, therefore process fault diagnosis is delayed, but not eliminated, by fault response based on symptoms.

The current state-of-the-art in process supervisory systems is to assist the operator to perform the function of fault *detection* through alarms for the most important and the most commonly encountered faults (Lees, 1983). Actions for the operator to take may then be prescribed in the form of alarm response procedures, usually a set of written procedures contained in a manual in the control room. Even when special care is taken to organize alarm responses carefully, the use of such procedures in off normal situations is inefficient.

Motivation for Use of Knowledge Based Systems

To be effective enough to be of use in the control room, a computer based diagnosis system must emulate the diagnostic powers of the best and most expert operators. Some of the characteristics of the best operators which such a system should emulate are given in Rasmussen (1974):

- The best operators understand what mode the process is in, and how the available measurements should look in that mode when the process is performing normally. Some variation in process measurements about a nominal value is expected due to process drifts and measurement noise, and these variations are not misinterpreted. From a quick scan of the process instrumentation panels, noting where gauges and dials lie with respect to where experience says they should, the best operators quickly gain a feeling of whether the process is behaving normally.

- The best operators know the manner of unit response to control actions involving setpoint changes, and recognize when an action of the automatic control system produces unusual or unexpected results. They also know the expected process response in certain common faulted situations, and from the pattern of that response are able to perform diagnosis quickly for these common faults and to take appropriate corrective action.

- The best operators have an internal model of the process and are able to call upon that model when faced with unfamiliar situations. The model is built upon an in-depth understanding of how each part of the process operates, and how the parts operate together to do their job.

In summary, a good operator has a "feel" for the process - where the values of the available measurements should normally reside, and what the derivatives of the measurements should be in controlled and common faulted situations. The heuristics of "feel", combined with deep qualitative and quantitative knowledge of process structure and behavior, give the basis for process fault detection and diagnostic systems.

The prerequisites for development of a knowledge based system are the accumulation of sufficient knowledge of process behavior and structure, and the representation of that knowledge in a form that a computer can understand. The application of knowledge based systems to continuous process fault detection and diagnosis has some unique problems:

- The information upon which the knowledge based system acts is not a static database, but is being continuously updated. This is because the situation for which reasoning is occurring is itself dynamic.

- All of the needed information must be gathered from process instrumentation; there can be no direct inquiry of the operator during a dynamic situation.

- Reasoning must proceed in the absence of needed information from unavailable or obviously faulted measurements.

- The system must be robust in the presence of multiple faults because processes may be operated in a degraded condition, and because propagation of fault effects through the process may create consequential problems.

- Knowledge of detailed quantitative process models is limited in many applications. Therefore the system must be structured to reason from qualitative knowledge where more quantitative knowledge is unavailable or uncertain.

- Results must be produced in a timely fashion - that is, soon enough so that an operator taking action on the basis of the results of the diagnostic system can have a significant mitigating effect and minimize economic and safety consequences.

Similar Work Reported in the Literature

A number of active researchers are working in this area. To summarize briefly some of the more widely reported efforts:

- A cooperative effort between the University of Delaware and DuPont produced a system called FALCON (Lamb, Chester, and Dhurjati, 1985). This was mainly a symptom based diagnostic system using expert system techniques commonly used in medical diagnosis.

- Work in diagnosis of nuclear station process faults is being investigated at Halden, Norway, under the sponsorship of OECD. This work consists of a rule based reasoner and a dynamic simulation model of the process. In developing the rules, an attempt is made to make explicit identification of the relationships between patterns of observed behavior and the presence of particular faults. Such features as temporal reasoning (reasoning about the sequence of occurrence of observations) and reasoning using certainty factors is included. Explicit fault response patterns are obtained from accident simulations and review of pass actual incidents reported in the literature. (Berg, et. al., 1985, Berg, et. al., 1987, Berg and Yokobayashi, 1986)

- NASA has sponsored process diagnostic research based on the need to analyze possible failures remotely without the ability to access the process for detailed troubleshooting. These methods have been mainly used in diagnosis of faults in ground support systems for the space shuttle. Knowledge representation is in the form of frames, with an explicit tie between frames to represent component connectivity. This type of knowledge representation allows a one-to-one correspondence between the frames and the process schematic diagram. (New, 1987)

- The Laboratory for AI Research at Ohio State University has created a high level AI building environment called CSRL. This tool is being applied to problems of diagnosis of process systems and sensor validation in nuclear stations and chemical plants. CSRL is being extended using a Diagnostic Applications Framework to increase transparency and portability of AI software and provide for improved maintainability. (Davis, et. al., 1985, Chandrasekaran and Miller, 1985)

- Several commercial AI shells which claim to provide an environment for the development of process diagnostic systems are being offered. These are powerful and complex software systems and are generally on the high end of the chart as far as cost and machine resources are concerned. These include G2 by GENSYM of Cambridge, MA, TESTBENCH by the Carnegie Group of Pittsburgh, and IDEA by *ai squared* of North Chelmsford, MA. Not as much technical information is available on these systems as for research systems, as one would suspect for competitive reasons, however, a cursory review of their advertised capabilities indicates that currently available versions are limited in their scope in terms of the forms of knowledge representation allowed and the general inferencing architecture.

- Process diagnostic work using knowledge based systems is being performed at the Laboratory for Intelligent Systems in Process Engineering (LISPE) at MIT's Department of Chemical Engineering (Kramer, 1987, Finch and Kramer, 1988, Oyeleye and Kramer, 1988). The work there is concerned with problems in the qualitative representation of continuous process data, model based reasoning in processes with high interconnection (such as processes with modern feedforward and feedback control), and practical means for delivery of the diagnostic system using PC based hardware and software systems.

Symptom Based Reasoning about Process Faults

As might be suspected, the earliest applications of knowledge based systems in process monitoring and diagnostics center around symptom classification systems, with the main form of knowledge representation being that of production rules. This occurred because of the precedent established for the success of such systems in earlier work on disease diagnosis and alarm tree based reasoning systems. A general discussion of the use of symptom based expert systems for diagnosis is contained in Mussalli and Fritsch (1986).

Basing process diagnosis on symptom based reasoning requires one to rely heavily on the ability to predict abnormal process behavior. The *a priori* knowledge needed - whether in the form of alarm trees, cause consequence diagrams, rule bases, or other response patterns - is generally developed by laboriously simulating process response to a large number of likely failures, then representing that abnormal behavior in an appropriate data structure against which a comparison may be made. The difficulties inherent in this method are discussed below.

- There is no way to anticipate every failure which may occur in a process. Of necessity, therefore, the amount of knowledge which can be assembled concerning the external indication of process failure is limited. If a failure is not represented explicitly in the rules, it cannot be recognized.

- It is difficult to predict using simulation the course of process response because of process non-linearities. The very conditions of most concern in predicting the behavior of the process under faulted conditions may violate the assumptions under which the simulation models were constructed in the first place.

- Such systems are ineffective at recognizing multiple faults, such as that created by an independent second fault or the occurrence of a fault in a process which is already in a degraded condition.

- Few symptom based reasoning systems can use dynamic information effectively, and then only by use of complex truth maintenance schemes.

- The most glaring deficiency of symptom based reasoning in diagnosis is the lack of facilities for explanation of root causes of process maloperation. Because these systems contain no deep process knowledge, neither of the specific process for which they are applied or of processes in general, explanation is restricted to repeating the imbedded relationship between symptom and cause. Such surface explanation is inadequate to ensure that proper understanding of the process fault has been achieved, and may mislead the operator in cases which are similar, but not identical to those upon which the heuristic knowledge base was constructed.

- The best predictive capability of preestablished pattern of fault response can be invalidated by an unanticipated operator intervention.

Beyond Symptom Based Diagnosis - the Use of Models

Improvement in fault detection and diagnostic system performance depends on the development of forms of knowledge representation which can capture deep knowledge of specific processes and process behavior in general. Such deep knowledge includes fundamental physical laws, causal connectivity relationships among the parts of a process system, functional as well as behavioral information concerning process equipment, and patterns derived from historical process behavior.

The following discussion will concentrate on those methods of knowledge representation which use some form of model; that is, a mathematical, geometric, or symbolic representation which captures and organizes deep process knowledge and can be used to predict behavior. A number of types of models have been proposed for use in process diagnostics, including a quantitative models based on modern control theory. However, unique modeling methods associated with the use of knowledge based reasoning systems have been developed. These models may be classified as follows:

- constraint based models - presentation of the model in the form of constraints, or mathematical relationships between parts of the process (states and parameters), which restrict behavior. In constraint based models, the rules may be derived from the constraints imposed on process behavior by the physical laws which govern the process.

- tree based models - these models attempt to represent faulted behavior by concentrating on connectivity relationships among the causes and effects of faults. This representation is generally in the form of a geometric structure relating the parts of the process to each other which defines paths of cause and effect for events which occur in the process. There are various forms of such models, including fault trees, event trees, goal trees, success trees, and response trees.

- primitive based models - in these models, a set of process primitives is used to break the process down into smaller standardized parts which represent fundamental models of process behavior. For a complex process, this implies that

higher level process functions are represented by a few lower level relationships derived from physical laws of process behavior.

- models based on formal systems of logic - in these models, the problem of process diagnostics is related to a formal logical system through isomorphism; that is, a mapping of the elements of process diagnostics to the symbols of a formal system of logic, such as predicate calculus.

- qualitative process models - in this form of representation, the quantitative details of process response are abstracted away and replaced by a qualitative representation. For example, each process measurement may be assigned a discrete range representing whether the value of the state is at its expected value or is higher or lower than expected. Fault detection is based only on the patterns which may be derived from the qualitative knowledge of the process, and not on the numerical values of the states.

The most effective diagnostic system will use a combination of knowledge representation methods, including symptom based rules, in order to perform in the most robust manner. In the work to be described below, qualitative representation of process response is combined with process causal models based on graphical representations of structure. The use of qualitative representations of process states by the mapping of process measurements to some form of quality space, combined with causal models which describe patterns of qualitative response, appear in the most promising of the recently proposed diagnostic systems. It has been recognized that humans are able to exist in the physical world and operate its processes very well without explicit knowledge of the specific differential equations which govern behavior of those processes. The task of the qualitative physics of a process is to derive its *behavior* from knowledge of its *structure*, but to avoid representation of the *function* in the description of the structure. This careful separation of the knowledge of structure, behavior, and function is the key to effective representation. A more complete discussion of these issues is contained in de Kleer and Brown (1984) and Kuipers (1984).

Knowledge Representation of Processes for Diagnostic Purposes

The following principles of process behavior apply to the methods of process knowledge representation to be used as a basis for a knowledge based diagnostic system:

- Fault response is local - this principle implies that the manifestation of a process fault shows its effect at the point at which the fault occurs. Thus faults can be identified and localized through a system of process measurements which are sufficiently sensitive and responsive, along with an organizational structure which permits mapping between individual measurements and their location in the process.

- Fault effects propagate through causal paths - the effects of faults propagate from the point of initiation through processes along predeterminable causal paths. These causal paths are governed by structural considerations, and may therefore be deduced from a representation of process structure. Part of this principle is the idea that faults must propagate (causal paths are inevitable), and the lack of propagation of a fault points to its invalidity.

- Faults are indicated by output-input inconsistency - the presence of a fault is indicated not by improper deviation in the indicated outputs of process blocks, but by an inconsistency between the actual response of a process and the expected response based on its inputs. This principle is very important in diagnostic reasoning, as for many faults a large number of measurements may show a deviation from their expected values. Diagnostic discrimination requires that the input-output inconsistency, rather that the presence of output deviation, be considered.

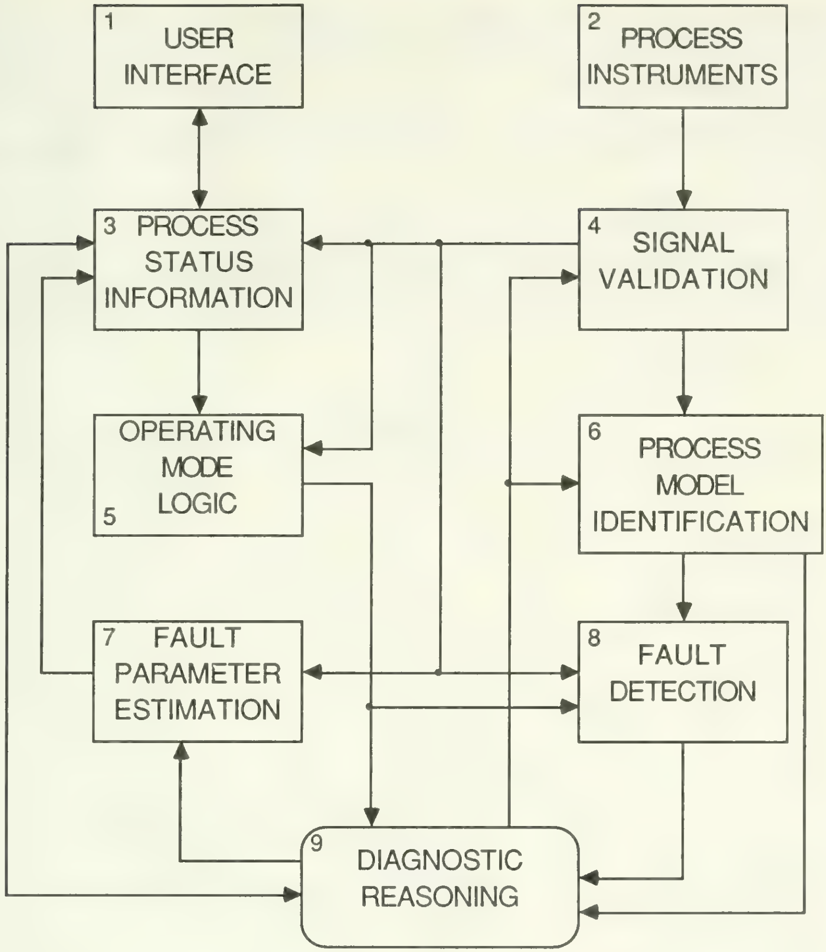


Figure 1. Knowledge Based Diagnostic System Functional Diagram

Knowledge Based Diagnostic System Design

There are four major functions to be performed by the process diagnostic system. These are gathering of process information from the data acquisition system, interpretation of the process information as a series of diagnostic events, reasoning about the events to determine the root causes of events, then providing an output to the user which provides proper information. The

functional diagram in Figure 1 represents the operation of the knowledge based diagnostic system. Each block is numbered to facilitate the discussion that follows.

Block 1 - User Interface

An effective interface would be required in a commercial product derived from the results presented herein, but the primary emphasis in the following discussion is on other blocks in the diagnostic system.

Block 2 - Process Instrumentation

It is assumed that the data to be furnished to the diagnostic system is acquired and stored by computer. This block represents the transfer of that data, in an appropriate format, to be read and processed by the diagnostic system. In this block, engineering unit conversions and normalizations will be performed, as required, in order to produce process measurement information upon which the diagnostic reasoning will act.

Block 3 - Process Status Information

In block diagram form, the process status information inputs and outputs are shown in Figure 2.

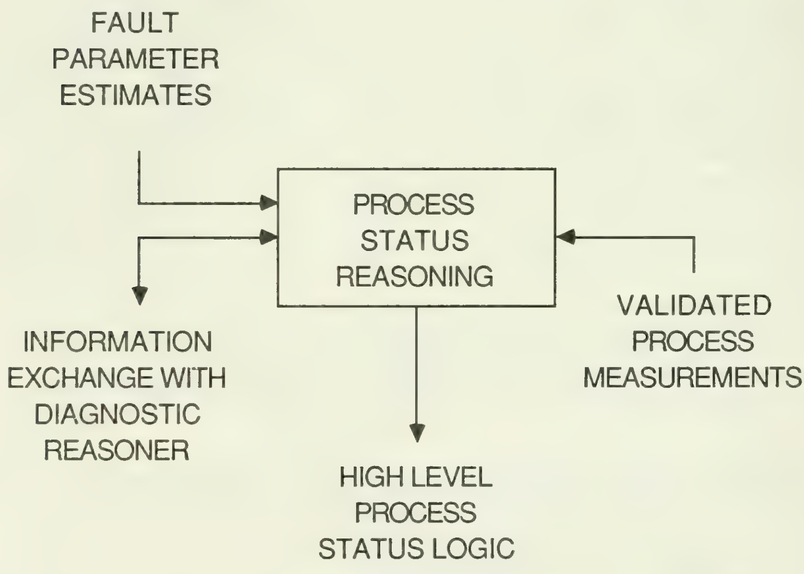


Figure 2. Process Status Information Block

In any practical implementation of the diagnostic system, a portion of the development of high level process knowledge would be done by the data acquisition system or supplied by the process

operator. Such information from the data acquisition system would include the overall process throughput (such as the megawatt output of a power station). Information suppled by the operator would include equipment out of service (when it affected what would be considered normal behavior of the process). This block represents the transfer of such high level process status information to the diagnostic system.

Block 4- Signal Validation

A wide variety of signal validation techniques are available. This block provides validation of redundant sensors using comparison or auctioneering techniques, and limit checking of all measurements and their trends. The purpose is to remove from further consideration those measurements which are obviously faulty, so that they do not trigger a diagnostic event. The block is shown in Figure 3.

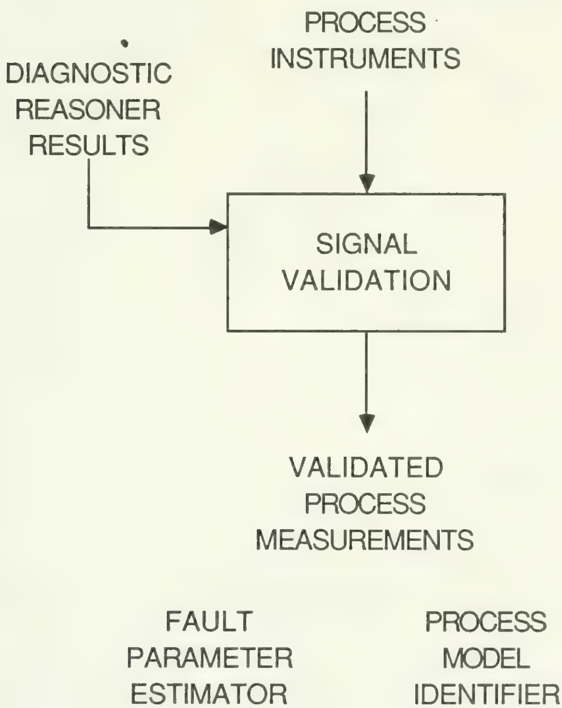


Figure 3. Signal Validation

Note that besides the absolute determination of measurement quality, this block receives an output from the diagnostic reasoning block. This input provides information to the signal validation module whenever the diagnostic reasoning determines that the only logical explanation for observed process behavior is a failed sensor, such as when a diagnostic event fails to propagate through the

process. The elimination of failed sensor signals by the signal validation block by either method enables diagnostic reasoning to proceed in the event of failed measurements.

Block 5- Mode Determination

The operating mode information is a summary logic for high level process status. It permits the mapping of process measurement values into a semantic description which may be used to assist the diagnostic reasoning in achieving robustness. The functioning of this block is shown in Figure 4.

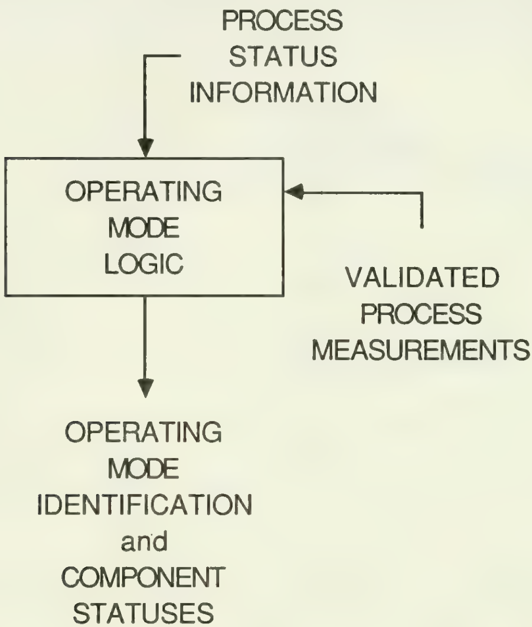


Figure 4. Operating Mode Determination

Process throughput or production rate is an important measurement because much of the later reasoning of the system concerning the interpretation of the various sensor signals will depend on the level. For example, each sensor signal is supplied a band within which it is considered to be "normal" which is in many cases a function of the level or production rate of the process. In conjunction with the process identification block described below, the mode and level determination performed by block 5 establishes the standard for the comparison of each instrumentation reading and thus the basis for reasoning using the qualitative causal model.

Block 6 - Process Model Identification

As noted in the discussion of block 5, it is necessary to define what values of measurements represent normal in order to identify potentially faulty behavior. For many measurements, the normal value is a constant, and the trigger levels for considering that measurement as indicating a fault may also be constants. However, in other cases, the normal band varies with production rate, equipment out-of-service, operating mode, and choices by the process operator. All of this diverse knowledge must be combined to produce the standard for process behavior against which comparisons for diagnostic purposes will be made. The functioning of this block is illustrated in Figure 5.

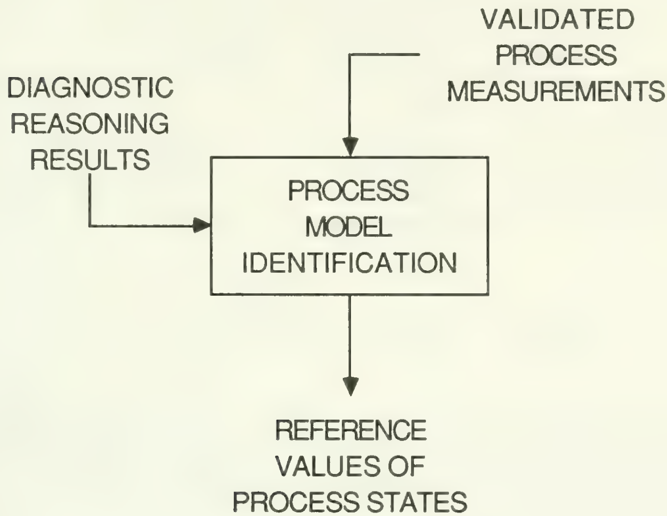


Figure 5. Process Model Identification

There are two methods by which information concerning the normal behavior of the process may be developed. The first is a deterministic method involving either curve fitting to process data gathered when the process is operating (similar to gain scheduling in adaptive control) or by a pattern matching expert system which derives its patterns from observations of normal operating data, similar to the principle involved in the System State Analyzer of Mott, King, and Radtke (1985). The second method uses an adaptive observer to estimate the transfer function of the measurements and then to predict the values of the measurements from the estimate of the model parameters. The methodology (recursive identification with exponential forgetting factors) is described in Ljung and Soderstrom (1983) and Landau (1976). In applying the latter method, the adaptation process is controlled by the diagnostic system; one example of the interaction between the qualitative and quantitative analysis methods used herein.

Block 7 - Fault Parameter Estimation

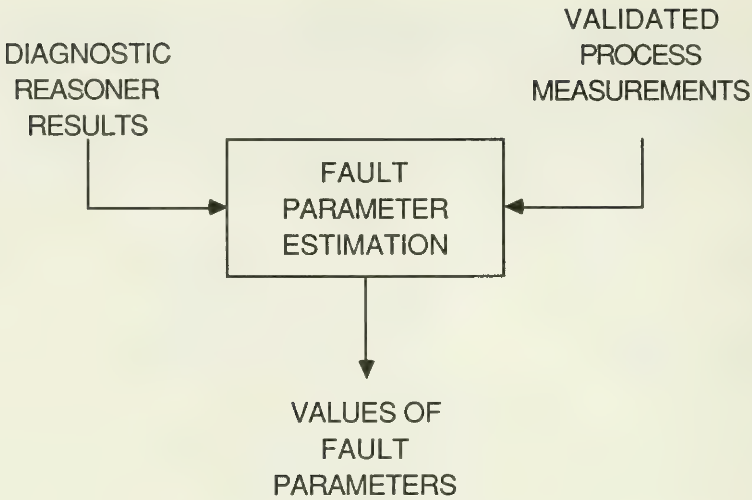


Figure 6. Fault Parameter Estimation

Once it has been determined that a fault is likely in the process, the block shown in Figure 6 uses numerical techniques to estimate the value of the faulted parameter(s) of the process. These parameters may include various flow resistances (blockages), leakage flows, heat transfer coefficients, etc. The estimates may then be used to formulate a strategy for correction of the fault be providing the operator with some indication of its severity and trend.

Block 8 - Fault Detection

Faults are detected when expected behavior deviates from observed behavior. A detected fault is called a diagnostic event, and there are many such events possible. If a measurement deviates from its normal value as determined by Block 6 above, then that is classified as a sensor based diagnostic event. If a process constraint is violated, such as an imbalance in mass flows around a volume, then that is classified as a constraint based diagnostic event. Events may also be potential (that is, expected but not yet observed) or consequential (occurring as the result of a previously detected event). The functioning of the fault detection block is shown in Figure 7.

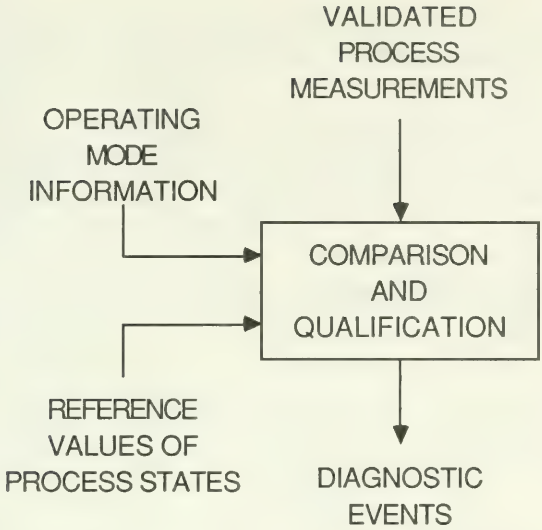


Figure 7. Qualification of Process States

Block 9 - Diagnostic Reasoning

The heart of the diagnostic system is shown in Figure 8, the diagnostic reasoner itself. The diagnostic reasoning block receives mode and level information, the qualitative values of the process measurements and constraints (diagnostic events), the covariances of the measurement estimators, and other process information which may assist in a specific diagnosis. Using qualitative causal models of the process, changes in the qualitative value of process measurements are examined. If these changes are unexplained by the known normal process response to operating level changes and measured disturbances, a fault is assumed to be occurring. Patterns of related measurements are examined to determine whether the fault is propagating in accordance with the known causal models of the process.

This block is also responsible for determining the root cause of the observed fault. It contains process knowledge about the relationship between qualitative values of process measurements and root causes of process failures. This knowledge includes fundamental process physics, such as the constraints imposed by mass and energy conservation, rule based reasoning using accumulated expert knowledge of process faulted response, such as that derived from records of previous failures, and generic failure information for particular process components which may be faulted. In the latter case, output from the diagnostic reasoner may activate further quantitative analysis of the process measurement information in order to estimate the fault parameters of these components.

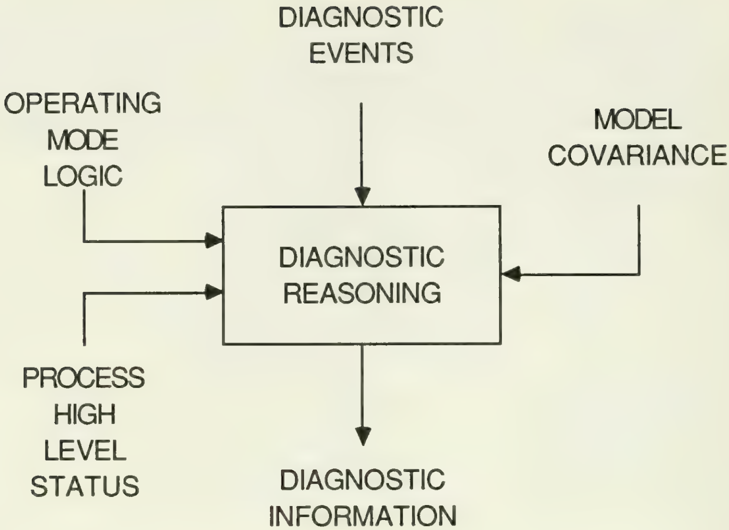


Figure 8. Diagnostic Reasoning

Application to Nuclear Station Diesel Engine Systems

In order to improve reliability and maintainability of the emergency diesel generators at the McGuire Nuclear Station, Duke Power Company, later joined by EPRI, undertook a project to improve the instrumentation of the engine and auxiliary systems. Data from this new instrumentation will be collected by a computer based data acquisition system and analyzed by an imbedded knowledge based system. The knowledge based diagnostic system design is as described in the previous sections of this report. The purpose of this section is to discuss the knowledge acquisition and analysis phases of the project.

Design information concerning the diesel engine and its auxiliaries was available because Duke Power had independently evaluated the original designs supplied by the engine vendor and had made modifications to better reflect the nuclear standby service that the systems would see. Complete failure histories of the McGuire engines and auxiliaries were available in the maintenance history records of the station. To add to our knowledge of diesel engine failures and causes, several additional documents were used.

- Previously prepared reports of diesel generator failure analysis - Diesel engine failures had been studied extensively by contractors to both NRC and EPRI (Driscoll, et. al., 1988, DeBey, 1988). These reports were studied as a basis for determining the most likely failures, because failures had been ranked in these reports by frequency of occurrence.

- Nuclear Plant Reliability Data System - By dumping all of the diesel and diesel auxiliary system failure information from NPRDS, a more complete picture of diesel failure could be developed. This database consisted of over 5000 failures in a number of different engines from various vendors. By downloading these files to a

PC based data analysis system, a number of studies could be performed. These consisted of charts of the various failures, the symptoms of the failures noted by the operators, and the corrective actions taken. These failures were analyzed on a system and component basis.

- Troubleshooting sections of a number of vendor manuals for diesel generator systems. These manual sections provided symptom based troubleshooting information for a number of observed operating failures, though it was apparent that most guidance was relatively general, and diesel generator systems generally lacked sufficient information to do comprehensive online troubleshooting of problems.

- Reliability Centered Maintenance study of the McGuire diesel systems - such a study is currently underway for the McGuire diesels and auxiliaries. A draft version of the report has been provided to the developers of the diagnostic system for information.

- Quantitative Model Studies - Dynamic models of the diesel auxiliary systems have been constructed using the Modular Modeling System (MMS). A FORTRAN model of the combustion process in a diesel cylinder has also been developed. Systematic studies of these models will provide additional verification of the diagnostic system by furnishing test data from simulations.

Based on the information gathered on various failures, a comprehensive list of symptom based troubleshooting information was compiled. This list consists of 24 categories of observed performance symptoms (such as "jacket water temperature"), with each of these categories having several possible observed faulted conditions, such as "high" and "low." There are several special categories of failures, such as exhaust color and running condition, with other forms of semantic descriptions for various faults. From these symptom categories, a rule-based expert system consisting of several hundred rules was constructed.

This rule base was constructed more as an informational exercise than as a specific approach to an embedded expert system. There were a number of observed problems with a rule based approach to diesel generator diagnostics:

- vague semantic descriptions of fault symptoms

- incomplete description of root causes

- lack of diagnostic discrimination, with each fault originating from a moderately large number of possible causes

- lack of mapping from semantic descriptions to specific instrumentation

However, the rule base is a useful tool to assist in the designation of the types and locations of new instrumentation to be installed on the engine. Our goal is to avoid a large amount of additional instruments, and to stay mainly with providing a computer based acquisition of existing system instrumentation. It is believed that this will minimize cost and complexity of the subsequent instrumentation installation. A second use for the rule base is to establish a minimum standard for the performance of the diagnostic system. We believe that the faults contained in the rules and troubleshooting charts should be found as unambiguously as possible by the diagnostic system using the installed instrumentation.

Applicability

There are three main areas of application for which the diesel generator diagnostic system is designed. The first involves the diagnosis of events which occur during the time that the engine is in standby, waiting for a start signal. The diagnostic system verifies that the engine is in a proper condition, including all normal temperatures, pressures, and levels, and that the starting air system is able to perform its function if required. The second area of applicability is the diagnosis of problems which occur during the startup sequence. The diagnostic system monitors the relay logic of the startup system, ensures that the starting automation has proceeded properly, and isolates failures in the starting sequence to a specific device or system. The third area is that of intelligent trending of diesel system measurements. This trending provides input for the maintenance of the diesel between test runs, and ensures that the engine condition remains proper for ensuring successful testing.

Conclusion

Work continues on the coding of the diesel generator system embedded diagnostic system. The schedule requires that a prototype be fielded later in 1989, and the system be complete by the beginning of 1990. It is expected that this system will improve the reliability of the engine and avoid having the diesel systems create limiting conditions for operation. We intend to extend this diagnostic methodology to other systems, particularly fossil fueled power stations, in conjunction with our program of control and monitoring system modernization in our power plants.

References

Berg, Ø., O. Evjen, U. S. Jorgensen, J. Kvalem, and I. Leikkonen (1985), "Early Fault Detection Using Process Models and Improved Presentation Techniques," OECD Halden Reactor Project Report HWR-141

Berg, Ø., R. E. Grini, T. Johansen, and M. Lilja (1987), "Early Fault Detection Demonstrated on the NORS Feedwater System," OECD Halden Reactor Project Report HWR-204

Berg, Ø. and M. Yokobayashi (1986), "Combination of Numeric and Symbolic Processing in Fault Detection and Diagnosis," OECD Halden Reactor Project Report No. HWR-174

Betancourt, J. M., D. L. Harmon, D. S. Jamison, A. Milliakos, and J. P. Rezendes (1984), *On-Line Success Path Monitoring: Aid to Restoring and Maintaining Plant Safety*, EPRI NP-3594

Chandrasekaran, B., and D. W. Miller (1985), "An Artificial Intelligence Approach to Sensor Conflict Detection, " *Proceedings of American Nuclear Society Topical Meeting on Computer Applications in Power Plant Operations and Control*, Pasco, WA

Davis, J. F., W. F. Punch, S. K. Shum, and B. Chandrasekaran (1985), "Application of Knowledge Based Systems for the Diagnosis of Operating Problems," presented at the AIChE Annual Meeting, Chicago

DeBey, T. M. (1988), *Predicting Emergency Diesel Performance*, SAND87-2473

de Kleer, J. and J. S. Brown (1985), "A Qualitative Physics Based on Confluences," published in Bobrow, D. G. (ed.), *Qualitative Reasoning About Physical Systems*, The MIT Press, Cambridge, MA

Driscoll, G. D., E. Oelkers, A. G. Pickett, and J. B. Redfield (1988), *Surveillance, Monitoring, and Diagnostic Techniques to Improve Diesel Generator Reliability*, EPRI NP-5924

Gaudio, P. J. and D. S. Jamison (1987), *Computerized Diagnostic Aid - Success Path Monitor*, EPRI NP-5088

Isermann, R. (1984), "Process Fault Detection Baed on Modeling and Estimation Methods - A Survey," *Automatica*, Vol. 20, No. 4, pp. 387-404

Kramer, M. A. (1986), "Integration of Hueristic and Model Based Inference in Chemical Process Fault Diagnostics," presented at the IFAC Workshop on Fault Detection and Safety in Chemical Plants, Kyoto, Japan

Kramer, M. A. and F. E. Finch (1987), "Fault Diagnosis of Chemical Processes," MIT Laboratory for Intelligent Systems in Process Engineering Report LISPE-87-019

Kuipers, B. (1984), "Commonsense Reasoning about Causality: Deriving Behavior from Structure," *Artificial Intelligence*, Vol. 24, pp. 169-203

Lamb, D., D. L. Chester, and P. Dhurjati (1985), "An Academic/Industry Project to Develop an Expert System for Chemical Process Fault Detection," Paper 70b, AIChE National Meeting, Chicago, IL

Landau, I. D. (1976), "Unbiased Recursive Identification Using Model Reference Techniques," *IEEE Transactions on Automatic Control*, Vol. AC-21, pp. 194-202

Lees, F. P. (1983), "Process Computer Alarm and Disturbance Analysis: Review of the State of the Art," *Computers and Chemical Engineering*, Vol. 7, No. 6, pp. 669-694

Ljung, L. and T. Soderstrom (1987), *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA

Mott, J., R. King, and W. Radtke (1985), "A Generalized System State Analyzer for Plant Surveillance," *Proceedings of American Nuclear Society Topical Meeting on Computer Applications in Power Plant Operations and Control*, Pasco, WA

Mussalli, Y. G., and T. J. Fritsch (1986), "Artificial Intelligence Techniques to Diagnose and Solve Heat Exchanger and Cooling Water Problems," *Proceedings of the American Power Conference*, pp. 551-556

New, E. (1987), "Knowledge Based Control and Redundancy Management Techniques Used in NASA's KATE Project, " Presented at Session 12, *Artificial Intelligence Applications for Industrial Control and Monitor Systems,* Southcon/87

Oyeleye, O. O. and M. A. Kramer (1988), "Qualitative Simulation of Chemical Process Systems: Steady State Analysis," *Journal of AIChE*, Vol. 34, No. 5

Rasmussen, J. (1974) , "On the Communication between Operators and Instrumentation in Automatic Process Plants, " contained in Edwards, E. , and F. P. Lees, *The Human Operator in Process Control*, Taylor and Francis, Ltd., London

Performance Diagnostic System for Emergency Diesel Generators

KEVIN P. LOGAN
MACSEA Ltd.
163 Water Street
Stonington, Connecticut 06378, USA

ABSTRACT

Diesel generators are commonly used for emergency backup power at nuclear stations. Emergency diesel generators (EDGs) are subject to both start-up and operating failures, due to infrequent and fast-start use. EDG reliability can be critical to plant safety, particularly when station blackout occurs.

This paper describes an expert diagnostic system designed to consistently evaluate the operating performance of diesel generators. The prototype system is comprised of a suite of sensor monitoring, cylinder combustion analyzing, and diagnostic workstation computers. "On-demand" assessments of generator and auxiliary equipment performance are provided along with color trend displays comparing measured performance to reference-normal conditions.

Abnormal conditions detected by the system produce diagnostic inferences on engine overloading, cylinder load balancing, and fuel injection timing. Performance fault prediction includes automatic trending functions which analyze time series of key measurements and performance parameters. The diagnostic strategies use a combination of rule-based, functional, and structural knowledge representations for automated reasoning.

INTRODUCTION

Station blackout is a recognized safety item in the nuclear power industry. Station blackout can occur when a plant trip coincides with the loss of both off-site and emergency backup power systems. Without emergency power, there may be limited or no means for reactor core cooling, increasing the likelihood of core meltdown.

Diesel generators are commonly used as the emergency power system. Emergency diesel generator (EDG) operating or start-up failure, subsequent to a loss of off-site power, can result in station blackout. The reliability of the EDG can therefore be critical to plant recovery and safety.

Beyond the safety issue, poor EDG reliability can result in significant costs to the utility. If the emergency power system fails, repairs must typically be completed within a week or less, after which time the plant must be shut down at a significant downtime expense.

A major expense in EDG operations is the requirement for a continuous scheduled maintenance testing program. Catastrophic failures in diesel engines are not uncommon. Relatively small deviations in critical engine operating parameters can result in component failure or thermal stress in critical subsystems. A comprehensive program of monitoring, diagnosis, and maintenance is required for reliable, economic EDG operation.

EDGs are typically tested at regular intervals, being started and operated at load conditions for a limited time period. Operating performance data is logged, evaluated, and used as a basis for engineering maintenance decision making. Plant engineers have varying levels of expertise in diesel generator operation and in their ability to evaluate and diagnose problems based on logged data. A lack of this expertise could result in undiagnosed generator problems, directly impacting EDG reliability. In addition, the generator performance evaluation may take a considerable amount of time by the plant engineer, the bulk of which is readily amenable to computer automation.

Automation of many diagnostic tasks through expert systems technology can provide a base of expertise in EDG performance assessment. This base may be progressively raised to new plateaus as expert systems are implemented and their knowledge bases extended.

An expert diagnostic system can introduce consistency in performance evaluations of EDGs, resulting in less reliance on the varying analytical skill levels of plant engineers.

The functions of an expert diagnostic system can be viewed in the context of EDG condition monitoring activities. The failure cycle of most equipment is described by a sequence of events that triggers maintenance action. This sequence includes fault detection, fault isolation or diagnosis, and fault repair or recovery. There is also the possibility of predicting a fault in advance based on a real-time analysis of plant performance parameters.

An expert diagnostic system can automate those functions associated with fault detection, fault diagnosis, and fault prediction. Through the use of on-line sensor measurements and a knowledgebase component which includes reference performance models for the EDG, fault detection and diagnosis can be accomplished through continuous comparisons of actual versus expected or reference-normal performance parameters. Fault prediction can be implemented through automated trending techniques which analyze time series of key EDG measurements and performance parameters. Such techniques can take advantage of the fact that before many equipments fail, they undergo a period of increasingly unstable or unreliable behavior. This behavior may be recognized as a time-dependent process which follows a particular trend model. Curve-fitting algorithms can be employed in the trend analysis to match the time series data to specific trend models. The trend models can then be used for fault prediction, based on an extrapolation of the trend over some future time interval.

Maintenance and repair costs may be reduced through a comprehensive scheduled maintenance program, coupled with the expert system for condition monitoring and diagnosis. The expert system may also serve to reduce preventive maintenance and surveillance requirements, along with their associated costs. An expert diagnostic system can help identify problem areas and alert engineers to take corrective action prior to component or subsystem failure. Such a system can provide engineers with diagnostic information for adjusting EDG operating parameters to within a specified tolerance of the reference-normal performance levels.

This paper describes a prototype diesel engine performance diagnostic system which has been under development for the past two years. The system is called the Diesel Expert Test Engineering Reasoner (DEXTER). DEXTER was designed for diagnosing performance problems on large marine diesel engines, but is just as applicable to the EDGs used in the nuclear power industry. This paper discusses the different knowledge representations and reasoning strategies being developed for the expert diagnostics. It also provides a detailed description of the prototype system, along with the mechanics of its operations.

Specific features of the prototype system include:

- Automated assessment of sensor instrumentation accuracy through built-in sensor validation procedures,

- Indication of deviations in diesel engine performance as compared to the manufacturer specified reference normal condition,

- Detection and diagnosis of causal factors of performance degradations, and

- Prediction of maintenance actions for improving performance and avoiding catastrophic engine failures.

KNOWLEDGE REPRESENTATIONS

Human knowledge assumes varying forms when transformed into computer knowledge. The structures by which knowledge is encoded into machine form have a significant effect in solving diagnostic problems.

Most current expert systems for diagnosis depend on compiled knowledge relating given symptoms to specific faults. The knowledge representation is comprised of sets of rules which are satisfied when specific patterns or combinations of symptoms are present. The initial prototype of DEXTER was developed along these lines. However, when new engine faults occur for which no diagnostic rule is satisfied, such as the case of only partial symptoms, the strict rule-based approach can be inadequate. Expert systems built from the rule-based approach alone may lack the robustness needed to diagnose new or unusual faults, even though partial evidence of a fault exists. Their reasoning mechanisms are based on inference rules following Boolean logic (i.e. combinations of AND and OR expressions which are tested for truth). The rule antecedents must be completely satisfied (i.e. logically true) in order for the rule to perform its intended diagnostic function.

Additional rules can be added to cover new situations arising from experience and the rule-base can continue to grow indefinitely as new discoveries are made. However, there may be a large number of symptom combinations which need to be included in the system, causing the knowledgebase to become excessively large. In so doing, it may also become less generalized to the domain topic and more specialized or engine-specific. In any event, the rule-based system will remain fragile in terms of its capabilities, due to the limited "surface" knowledge representation associated with rules.

The diagnostic capabilities of the expert system can be extended by using other types of knowledge representation and reasoning strategies which can generalize from partial symptom information. Other types of knowledge include time-based or temporal trend information, structural models of the engine systems, and maintenance related information on overhauls and repairs of engine components.

An expert system with robust diagnostic capabilities requires knowledge representation and reasoning strategies that go beyond simple rule-based logic. Figure 1 illustrates the types of knowledge designed into DEXTER, including functional, experiential, structural, and temporal representations.

Functional Knowledge

Functional knowledge involves system models describing correct, unfaulted engine performance. Diagnostic reasoning is based on discrepancies between expected performance, as derived from the models, and measured engine performance.

DEXTER represents functional knowledge by relationships among key engine performance parameters, defined empirically through the testbed trials. Expected engine behavior is determined from testbed relationships, such as those shown in figure 2.

Functional knowledge is used to generate a set of qualitative engine performance parameters, based on comparisons between observed and expected or reference-normal conditions. Performance deviations from reference conditions are transformed into discrete qualitative measures having HIGH, LOW, or NORMAL values.

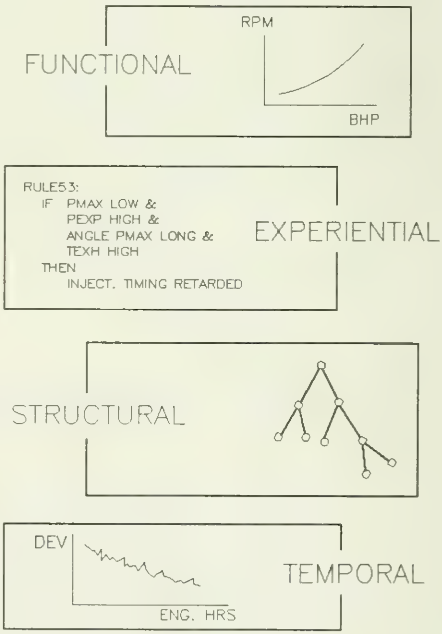


Figure 1 - Knowledge Representations within DEXTER

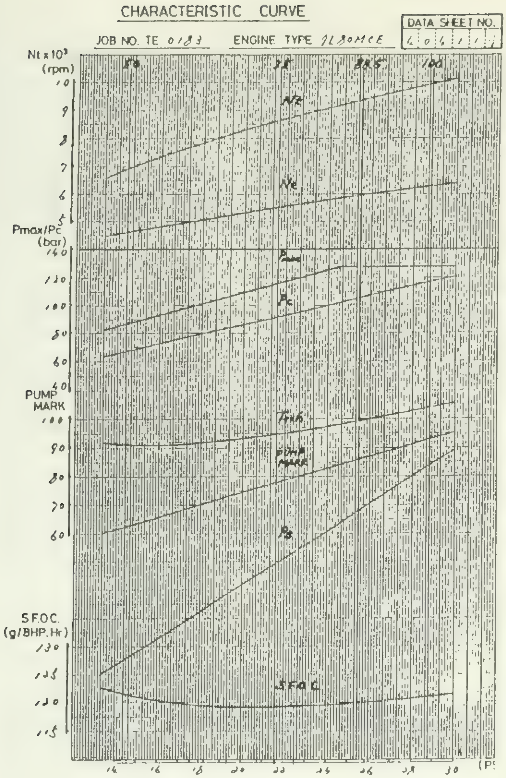


Figure 2 - Engine Testbed Data

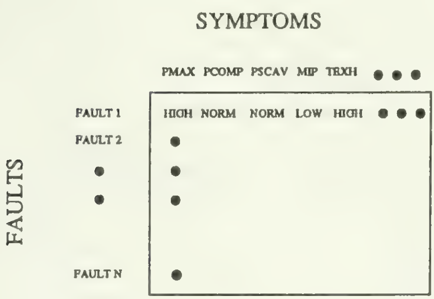


Figure 3 - Fault/Symptom Matrix

Experiential Knowledge

Experiential knowledge defines expected engine fault behavior based on the accumulated experience of experts. This form of compiled knowledge maps a set of possible engine faults to their matching symptoms. Rules associate specific symptoms to possible faults. The rules are equivalent to fault models of malfunctioning system performance, often depicted as fault trees or fault-symptom matrices.

Fault hypothesis generation relies on the compiled knowledgebase containing descriptions of all possible faults and their associated measurable symptoms. Diagnostic reasoning is restricted to pattern matching of observed symptoms to prestored or compiled symptoms. A match between the observed and the compiled symptoms generates fault diagnostics in a manner analogous to record retrieval in an indexed database. The database in this case consists of a rule knowledgebase of IF/THEN type relationships or an equivalent fault-symptom matrix, as depicted in figure 3.

Rulebased diagnostics work well if there is a one-to-one mapping of symptom patterns to system fault conditions. In this case, a given set of symptoms will be deterministic in diagnosing the correct fault. However, if this is not the case, a given symptom pattern may indicate that any one of several possible system fault conditions exist, resulting in the generation of multiple fault hypotheses.

Structural Knowledge

Structural knowledge representation involves a hierarchical description of the device under diagnosis in terms of its basic components and their connectivity. The connectivity may be defined in several ways, such as physically, relationally or causally, and functionally. A given device is comprised of various subsystems or components, which can themselves be described in terms of their components. The hierarchical breakdown of device components and interconnections need only be carried to a level of detail corresponding to the least replaceable components of the device.

Diagnostic reasoning based on device structure can be used to isolate a candidate set of faulty components. Structural knowledge allows backtracking along a path of connectivity to identify all components associated with a known abnormal symptom. Sensory information, associated with specific devices or components, is input to functional models to determine expected performance behavior. When the output deviates from expected behavior, the candidate set of possible faulty components is formed from the associated connectivity paths.

Structural knowledge can be used to extend rule-based diagnostics. Structural information can isolate suspect components at the lowest levels of the structural hierarchy for which there are no measurements and/or reference conditions of expected behavior.

Temporal Knowledge

One form of temporal knowledge defines dynamic or time-dependent changes in engine performance parameters. Performance often slowly degrades to an abnormal state. Degradation trends provide supporting evidence for specific faults and contradictory evidence against other faults. This knowledge helps to identify the correct fault when several possibilities exist.

Another form of temporal or time-dependent knowledge involves historical maintenance information. Historical maintenance records and component specifications can be used to determine the most likely faults within a candidate set of components. Historical records provide information on when specific components were last overhauled. Component specifications indicate expected time between overhauls or mean time between failure.

SYSTEM ARCHITECTURE

Figure 4 illustrates the overall design of DEXTER. The bold outlined boxes represent system modules developed during the current research phase.

Data Acquisition Module

Plant performance data is supplied to the system under the control of the data acquisition module. This includes software for data communications with the engine monitoring and cylinder combustion analyzing computers, as well as special processing routines for measurement inputs.

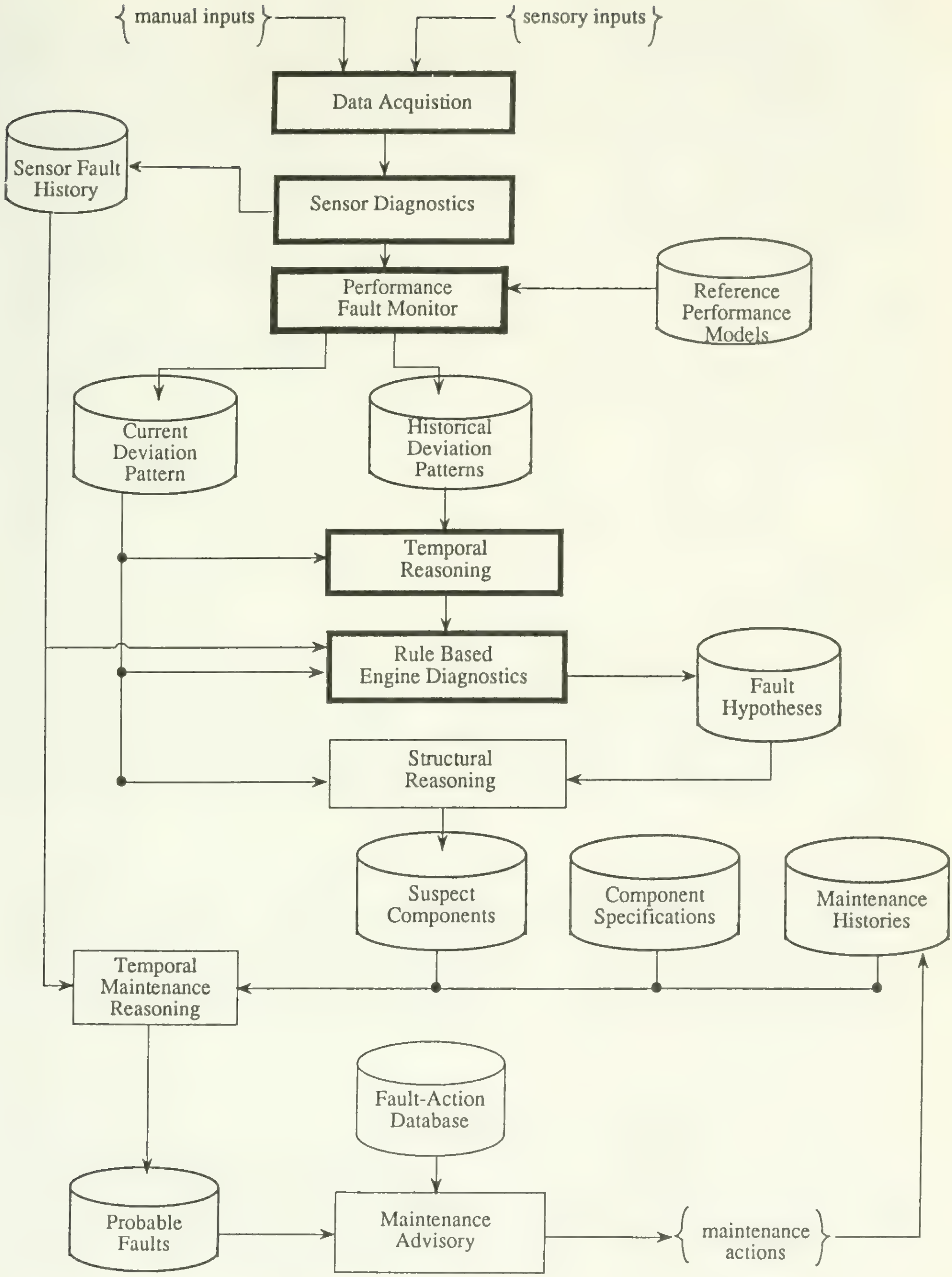


Figure 4 - Overall DEXTER Design

Sensor Diagnostics Module

Consistency checks are applied to selected measurements supplied by the data acquisition module. Consistency is established on the basis of both direct and analytically redundant measurements of engine parameters. Sensor faults are identified and validated estimates of engine parameters are computed. Sensor fault information is archived into the knowledgebase.

Performance Fault Monitor

Using the input set of engine parameters, reference performance parameters are established and compared to operating data. A current deviation pattern is formed by transforming parameter deviations into qualitative measurements reflecting normal or abnormal behavior. The current deviation pattern is archived into the knowledgebase to record a history of such patterns.

Temporal Reasoning

Historical deviation patterns are examined for significant trends over pre-established intervals of engine running hours. Trend information is generated on each performance parameter for both short-term and long-term intervals.

Rule-based Diagnostic Reasoning

Current and historical deviation patterns, as well as trending results, are used to diagnose possible engine performance problems. Reasoning is based on a compiled knowledgebase of rules. Historical trending information lends evidential support to diagnostics based on the current deviation pattern. The result of the rule-based reasoning is a candidate set of fault hypotheses. When this set includes multiple hypotheses, fault resolution proceeds through the structural reasoning module.

Structural Reasoning

The set of candidate faults is expanded through a structural enumeration of possible fault paths connected to the identified components. A list of suspect components is generated whose malfunction describes the observed performance of the engine. The suspect components are those occurring along paths of structural connectivity with those components associated with the initial set of fault hypotheses.

Temporal Maintenance Reasoning

Given an input set of suspect components, this module assesses historical maintenance records and component specifications to determine the most probable faults within the candidate set. Historical records provide information on when specific components were last replaced, repaired, or inspected, based on engine running hours. Component specifications provide information on expected time between

overhauls or mean time between failure. Based on this information, the suspect component list is used to statistically derive the most likely failure candidates.

Maintenance Advisory

The results of temporal maintenance reasoning are output to the user through the maintenance advisory module. The user is able to inquire about supporting and contradictory evidence, as well as paths of reasoning, for each conclusion reached by the system. The advisory function can be extended to provide maintenance instructions associated with each maintenance activity.

Maintenance Activity Feedback

As maintenance activities are performed, the maintenance history database is updated to reflect recent repairs and inspections. This provides a feedback loop of information flow for subsequent cycles in the diagnostic process.

REASONING STRATEGIES

DEXTER'S design architecture allows combining different types of reasoning strategies to continually refine engine fault hypotheses. Different knowledge representations contribute to the formation and resolution of one or more fault hypotheses.

Temporal Reasoning from Performance Trends

Temporal reasoning involves examining information for all the engine parameters and factoring any significant trends found into the diagnostic reasoning process. Trend information is generated through special trending routines, which are called from the temporal reasoning module.

Temporal reasoning lends contradictory or evidential support to fault hypotheses generated from simple rule-based diagnostics. When multiple hypotheses exist, temporal information can help assign probabilities to certain faults on the basis of historical performance deviation trends appearing in the data. When components fail, they typically experience slow degradation rather than catastrophic failure. The presence of such trends will lend support for, while the absence of such trends will lend support against, given fault hypotheses.

Temporal reasoning can be implemented by maintaining time history information of predefined length for each system parameter. Regression analysis is used by the trending function to quantify the statistical relationships present in the data. These relationships can be transformed into qualitative or symbolic attributes of time-based behavior, having values such as CONSTANT, INCREASING, DECREASING or UNKNOWN.

Rulebased Diagnostic Reasoning

Figure 5 conceptually illustrates rulebased diagnostics from snapshot engine data. The engine evaluation procedures are first applied to service performance data. Reference performance "models" are used to generate diagnostic patterns of deviations from reference normal conditions. The resulting diagnostic pattern is used in conjunction with historical fault patterns to infer fault conditions regarding engine subsystems and components. Alternative fault hypotheses are generated on the basis of these inference rules.

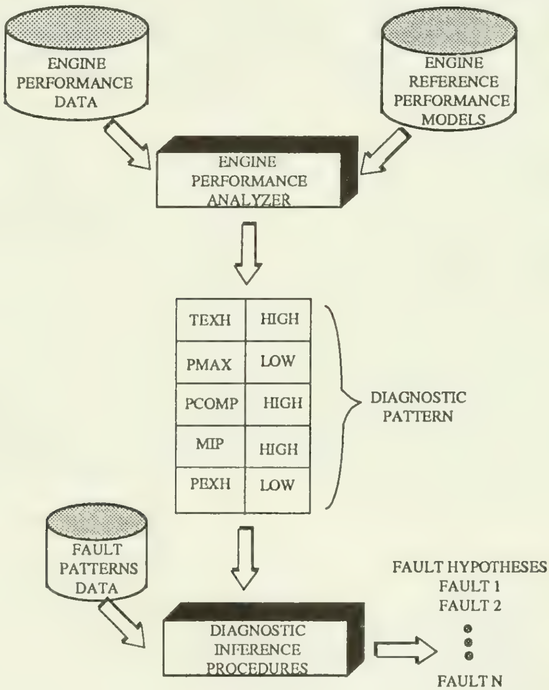


Figure 5 - Rulebased Diagnostic Reasoning

Structural Reasoning

The encoding of a device structural model for computer implementation involves the generation of a set of link descriptions. A link connects two device components and is described by a predecessor/successor pair. The ordering of the predecessor and successor components implies a control dependency used for fault path isolation. The correct operation of certain components is dependent on the correct operation of one or more other components to which it is connected.

Diagnostic reasoning based on device structure can be used to isolate a candidate set of faulty components. Structural knowledge allows backtracking along a path of connectivity to identify all components associated with a known abnormal symptom.

Computer programs have been developed to identify the connectivity paths from a given component to another within a specified structural model. The programs can also enumerate all possible paths of connectivity from or to a given component, thereby providing a tracing of potential causal fault paths. Using such programs, device structural models can be applied for diagnostic tasks to systems of arbitrary complexity. Hierarchies of structural models can be developed to describe subsystems and their components to any desired level of detail.

Current development work involves enhancing the path following program to use other types of component information, such as component reliability data, to determine most probable fault paths within a given structural model.

The structural models include descriptions of each component and its relationship with other directly connected components. Elements of the component's frame definition include the following attributes:

- name,
- controlling devices,
- input connection components,
- output connection components,
- failure probability, and
- descriptive information.

Temporal Reasoning from Maintenance Events

A major goal of any diagnostic system is to identify a failure (or possibly multiple failures) on the basis of all known information. This information can include symptoms derived from measurements and system parameters compared against expectations. It can also include information regarding the past history of the equipment, such as when maintenance actions were last performed. The latter type of information is typically stored in maintenance log files and updated periodically as maintenance actions are completed.

When engine performance faults are detected, a search for correlations between the symptoms and the maintenance histories of related or connected components may result in a more precise isolation of the most probable failure suspects. The maintenance histories of engine components can be used to develop statistical estimates of failure rates. In addition to the statistically derived failure rates, other

component specifications, such as date of last maintenance overhaul, repair, or replacement, and mean time between overhaul, can be used to statistically derive component failure probabilities as a function of engine running hours. Component failure probabilities are developed using standard techniques of reliability analysis.

Structural diagnosis can be extended by assigning these probabilities of component faults to the path links of the structural models. An ordering of the most likely fault paths can then be generated, using the link/component probabilities. In this way, experience involving component failures, compiled from machinery maintenance histories, is used to direct the focus of the system towards the more likely suspects when faults are observed. The expert system thus becomes cognizant of the history of component failures and appropriately factors this experience into the diagnostic process.

PROTOTYPE SYSTEM DESCRIPTION

Test Engine

The test engine, installed aboard the President Harding, is a low speed, two cycle, turbo-charged Mitsui-B&W 9L80MCE marine diesel engine. The maximum continuous rating for the engine is 28,800 BPH at 83 revolutions per minute. The engine has nine cylinders with a single fuel pump and two fuel injectors per cylinder. The engine also has two sets of turbochargers and air coolers, and a shaft generator attached.

Target Diagnostics and Measurements

The engine manufacturer's recommended performance evaluation procedures [1] established a target set of diagnostics for the major engine subsystems and components. These diagnostics cover the salient aspects of diesel engine operation, including the main engine and cylinder condition, turbocharger performance, and air cooler performance.

The initial target diagnostics address engine overloading, even power distribution amongst cylinders, fuel injection timing, turbocharger turbine and compressor fouling, intake air filter fouling, and air cooler fouling.

The target diagnostics establish certain measurement requirements and computed parameters. Table I lists the set of measurements required for the initial target diagnostics.

Major System Modules

The major program modules comprising DEXTER include data acquisition, sensor diagnostics, engine performance and trending analysis, and engine diagnostics.

Data Acquisition Module. The data acquisition module is responsible for obtaining all inputs to the system. The majority of the data elements are acquired through digital interfaces with the engine monitor and combustion analyzer computers, as depicted in figure 6. The design of the data acquisition module provides for "on-demand" data acquisition under user control. The user initiates data acquisition by selecting menu options provided on a CRT screen, which trigger data communications with the engine monitor and combustion analyzer. A capability is also provided for manually entering values through the system keyboard.

Through two customized interfaces, DEXTER performs data communications with the engine monitoring and the combustion analyzer computers. Engine measurement data is obtained on-demand by DEXTER in digital form. Both interfaces use RS-232 ports and special communications software.

Table I
Engine Performance Measurements

Engine running hours
Engine rpm
Engine power output
Shaft generator power output
Maximum combustion pressure per cylinder
Compression pressure per cylinder
Mean indicated pressure per cylinder
Exhaust gas temp. per cylinder
Scavenge air pressure
Scavenge air temp.
Turbocharger intake air temp.
Pressure drop across air filter
Turbocharger exhaust gas inlet temp.
Exhaust gas pressure after turbocharger
Exhaust gas receiver pressure
Pressure drop across air cooler
Air cooler water inlet temp.
Air cooler water outlet temp.
Fuel consumption
Specific gravity of fuel
Sulfer content of fuel
Fuel inlet temp.

Sensor Diagnostics Module. DEXTER's knowledge of engine condition consists of information from sensory inputs. From these inputs, DEXTER attempts to infer discrepancies from reference normal conditions and to diagnose the causal factors of off-design performance. Deviations in engine performance can generally be classified as:

1) A true degradation or failure in one or more engine components or subsystems,

2) A change in the operating environment, such as ambient conditions or fuel quality, or

3) A failure in one or more sensors.

The latter cause is addressed by sensor diagnostic techniques.

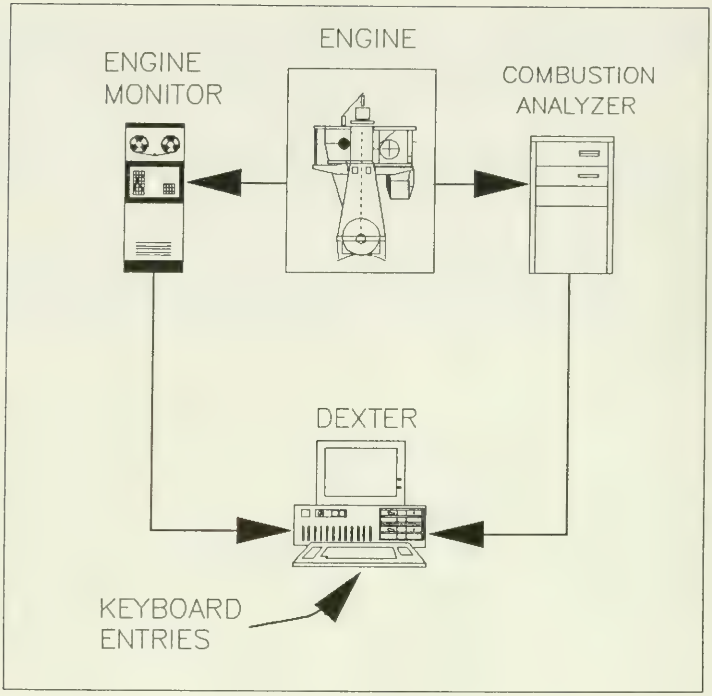


Figure 6 - DEXTER Data Acquisition Configuration

Sensor Diagnostic Technology. Various software techniques for sensor fault isolation have been proven successful in aerospace and nuclear applications [2-9]. Among the more successful techniques are parity-space representation and analytical redundancy. These techniques are jointly used to isolate failed sensors and eliminate their use in critical system computations. The techniques implemented within DEXTER are described in Appendix I.

Validation of Engine Power Measurements. DEXTER's engine diagnostic capabilities are dependent on an accurate estimate of engine power. The reference values for several key performance parameters are determined from testbed data on the basis of brake horsepower (BHP). An inaccurate estimate of BHP will result in inaccurate reference values, leading to false diagnoses of engine problems.

Sensor validation techniques were applied to engine BHP estimation, following the algorithm depicted in figure 7. The sensor validation model involves three analytical measurements of BHP. The first analytical measurement computes BHP from the torsionmeter SHP measurement and shaft generator load. The second estimates BHP from average turbocharger RPM, scavenge air temperature, inlet air temperature, and barometric pressure. The third analytical measurement is based on average cylinder MIP and engine RPM.

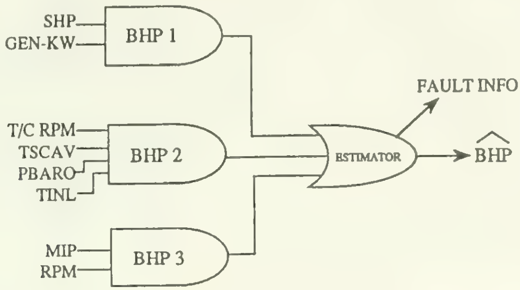


Figure 7 - BHP Signal Validation Algorithm

The algorithm was tested on engine performance data collected aboard the President Harding during the fall of 1988. Figure 8 shows a plot of the three analytical measurements of BHP, along with a plot of the validated BHP estimate. The weighted estimate was made with the error bounds set to 2 percent of MCR for each of the three analytical BHP inputs. The benefit of the sensor validation technique is demonstrated for the sample at 10,341 engine hours, shown as point A. Here, the combustion analyzer BHP estimate is inconsistently low compared to the other two estimates. The program flagged this input as faulty and excluded it from the calculation of the validated estimate. Further tests of the BHP signal validation algorithm showed it to perform well in the presence of corrupted data.

Engine Performance Analysis Module. DEXTER includes functions which automate the performance evaluation procedures set forth by the engine manufacturer [1]. Computations of fourteen main performance parameters, as listed in Table II, provide diagnostic information on the condition of the main engine, turbochargers, air filters, and air coolers. Certain parameters are corrected to standard reference conditions before being used to determine their corresponding expected values. The expected or reference performance values are determined from engine testbed trials, which are also corrected to the same reference conditions. The program generates graphic displays comparing each measured parameter against its corresponding reference value, as shown in figure 9. A record of all performance data is stored to computer file. This file is used to analyze time-based deviations in the individual parameters.

Table II

Engine Performance Parameters

Average exhaust gas temp.
Specific fuel oil consumption
Average fuel pump index
Average cylinder PMAX
Average cylinder PCOMP
Scavenge air pressure
Turbocharger rpm
Pressure drop across intake air filters
Turbocharger compressor efficiency
Turbocharger turbine efficiency
Temp. diff. - cooler air out-water in
Temp. diff. water across air cooler
Pressure drop across air cooler

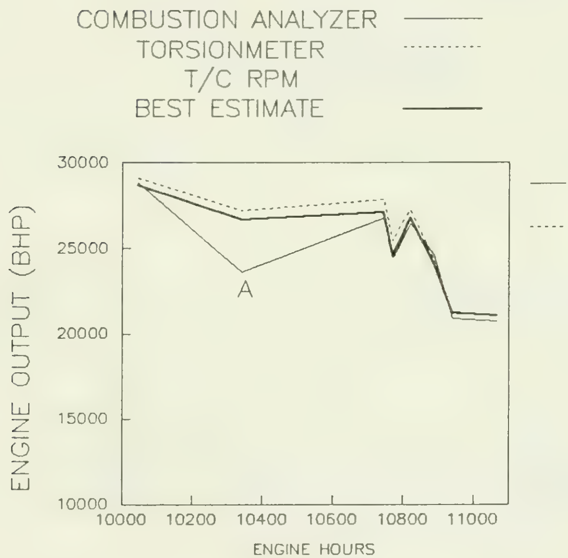


Figure 8 - Sensor Validation Results from Test Engine

Corrections for Ambient Conditions. Certain engine performance parameters are corrected to standard reference conditions before being compared to past and testbed values. PMAX, Texh, PCOMP, and PSCAV are corrected to reference conditions of 27 ^{o}C and 39 ^{o}C for air inlet and scavenge air temperatures, respectively. The correction procedures are applied to the reference testbed data, as well as to measured values, in order to put all data on a common basis.

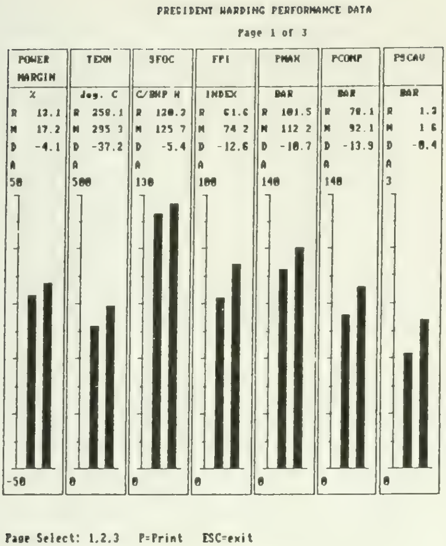


Figure 9 - DEXTER Performance Graph

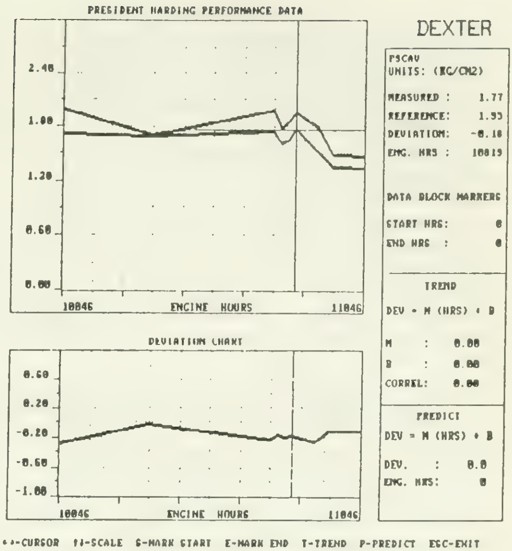


Figure 10 - DEXTER Trend Graph

The average fuel pump index is also corrected to the above temperature conditions, as well as for fuel calorific value and density. Specific fuel oil consumption is corrected similarly.

Trending Analysis Module. Degrading performance trends can be quantified through trend analysis. DEXTER uses regression analysis techniques to estimate the unknown parameters of linear, time-dependent models. Separate trend models are established for each key engine parameter. Information generated by the trending function is used in DEXTER's diagnostic functions.

Performance deviation trends indicate which engine component should be overhauled. The slope of the trend curve indicates when performance will reach a specified deviation threshold, at which time the overhaul should be carried out. This type of predictive capability was implemented as a user-interactive feature. The interactive trending function generates trend graphs of selected engine parameters on the CRT screen, showing time-based deviations against engine running hours, as illustrated in figure 10.

Trend data is displayed across a variable base of engine running hours. A function is provided to mark any block of deviation data for regression analysis. The data block is input to a regression function, which determines the statistical relationship between the selected parameter and engine running hours. The correlation coefficient, indicating the strength of the relationship, and the regression equation are automatically computed.

The prediction function uses the computed regression equation and a user-entered deviation limit to compute the engine running hours at which the limit value will be reached. This allows the engineer to determine when the engine parameter will exceed some critical value, necessitating corrective maintenance.

CONCLUSIONS

Emergency diesel generators are subject to operating failures due to fast starting and infrequent, off-design operation. In order to achieve high EDG reliability, utility companies must adhere to a continuous scheduled maintenance and testing program. A lack of experience and expertise in evaluating EDG performance test data can result in undiagnosed EDG operating problems, resulting in decreased reliability.

Automation of many diagnostic tasks through expert systems technology can provide a base of expertise in EDG performance assessment. An expert diagnostic system can introduce accuracy and consistency in performance evaluations. Such a system can help identify problem areas and alert engineers to take corrective action prior to generator failure. Diagnostic information can also be used to maintain EDG performance at reference-normal or design levels.

Most current expert systems are built from heuristic rules. Expert diagnostic systems which are strictly rule-based may lack the capability to diagnose new or unusual faults. A system with robust diagnostic capabilities requires knowledge representations and reasoning strategies that go beyond simple rule-based logic.

This paper has described an expert system under development which employs a combination of functional, experiential, structural, and temporal knowledge for diagnostic reasoning about engine performance. An important related area of development involves sensor diagnostics. The prototype system incorporates advanced signal validation techniques used in other nuclear applications, such as validation of Safety Parameter Display System signals.

The resulting diagnostic system will provide a base level of consistency in EDG performance monitoring, with less reliance on the varying analytical skills of plant engineers.

Practical benefits expected from the use of the expert system are:

- Indication of deviations in overall EDG and subsystem/component performance from the manufacturer specified reference normal condition,
- Reduction in the occurrence of off-design operation, thereby improving individual component and overall system reliability,
- Detection and diagnosis of EDG performance faults,
- Prediction of specific maintenance actions which will improve performance and avoid catastrophic failures.

ACKNOWLEDGEMENTS

This work was funded as part of a cooperative government-industry research program involving the U.S. Department of Transportation - Maritime Administration, American President Lines of Oakland, California, and several other industrial participants. The author gratefully acknowledges Mr. John Dumbleton of the Technology Assessment Office of the Maritime Administration, Mr. Doug Ward of American President Lines, American M.A.N. Corporation, and Veritas Petroleum Services.

REFERENCES

1. Instructions for L-MC/MCE Type Engines Operation, Edition 38, M.A.N. -B&W Diesel A/S, Copenhagen, Denmark.

2. J.J.Deyst, Jr., R.M. Kanazawa, and J.P. Pasquenza, "Sensor Validation: A Method to Enhance the Quality of the Man/Machine Interface in Nuclear Power Stations," IEEE Nuclear Power Systems Symposium, Orlando, Florida, November 5-7, 1980.

3. J.J. Deyst and J.C. Deckert, "Maximum Likelihood Failure Identification Techniques Applied to the Shuttle RCS Jets." AIAA J. Spacecraft and Rockets, Vol. 13, No. 2, February 1976.

4. K.C. Daly, E. Gai, and J.V. Harrison, "Generalized Likelihood Test for FDI in Redundant Sensor Configurations," AIAA J. Guidance and Control, Vol. 2, No. 1, January-February 1979.

5. M. Desai and A. Ray, "A Fault Detection and Isolation Methodology," Proceedings of the 20th IEEE Conference on Decision and Control, San Diego, CA, December 1981.

6. C. H. Meijer, J.P. Pasquenza, J.C. Deckert, J.L. Fisher, D.B. Laning and A. Ray, "On-Line Power Plant Signal Validation Technique Utilizing Parity Space Representation and Analytic Redundancy," Report NP-2110, Electric Power Research Institute, Palo Alto, CA, November 1981.

7. J. Fisher, O. Deutsch, A. Ray, R. Ornedo, E. Kujawski, C. Sawyer and P. Smith, "Demonstration of a BWR Suppression Pool Signal-Validation Program," Report NP-3641, Electric Power Research Institute, Palo Alto, CA, July 1984.

8. O. Deutsch, "Fault-Tolerant Systems Technology: Signal Validation," Trans. Am. Nucl. Soc. 49, 1985.

9. B.J. Benedict, S.S. Godbole, R.I. Lutz, and O.L. Deutsch, "Validation and Integration of Critical PWR Signals for Safety Parameter Display Systems," Report NP-4566, Electric Power Research Institute, Palo Alto, CA, May 1986.

10. P.A. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall International, 1982.

APPENDIX I

SENSOR DIAGNOSTICS USING PARITY SPACE REPRESENTATION AND ANALYTICAL REDUNDANCY TECHNIQUES

Parity-space techniques [6] transform a set of redundant measurements of a system variable into a new set of error values, represented as a parity vector. This transformation generates residual components of measurement errors as elements of the parity vector. The parity vector is analyzed to isolate sensor faults. When a fault occurs, the parity vector grows in magnitude and its direction of growth is uniquely associated with the faulty measurement. The parity vector behavior is an indication of both the presence of a fault and the identity of the faulty measurement.

Analytical redundancy involves using physical relationships among measurement variables to calculate other related variables. Analytic measurements allow failure of one or more direct measurements to be isolated via consistency metrics.

A sensor validation algorithm performs both fault isolation and parameter estimation. It is comprised of analytic measurement calculators and estimators. The design of the sensor validation algorithm includes an estimator for each key variable, with a validated estimate of the variable as output. An adequate number of redundant measurements (direct and analytic) are required for each critical variable. Two redundant measurements allow a validated estimate of a key variable to be computed, but sensor failure in this case cannot be determined. At least three measurements are required for a validated estimate in the presence of a single sensor failure. More redundant input measurements increases the likelihood of an accurate validated estimate of the key variable and also allows detection of common-mode sensor failures.

The sensor fault isolation methodology implemented within the estimator is based on recognition of inconsistencies among all redundant measurements, both direct and analytic. For any pair of measurements, m1 and m2, the magnitude of the parity vector is directly proportional to $|m1-m2|$. This quantity can be compared to the sum of the respective error bounds, (b1 + b2), for a consistency check. In this manner, the relative consistencies of scalar measurements can be evaluated by concurrent checking of all $n(n-1)/2$ pairs. A consistency metric, I, similar to the nonlinear and Minkowski distance metrics used in pattern classification [10], indicates the degree of consistency of each measurement with all other measurements.

For each measurement, the consistency metric I produces an ordinally scaled integer in the range (Ø, n-1), providing n different levels of classification against the measurement sample. Those measurements having the greatest consistency across the sample are least likely to have failed. Conversely, those having the least consistency are most likely to have failed.

The consistency metric is employed in the estimation of the process variable. For any measurement sample, an estimate of the process variable is calculated from a weighted average of consistent measurements. The weighting factors are determined by inverse variance weighting.

Expert System Monitoring Electric Power Plant Supplies—Bugey Nuclear Power Plant

J. ANCELIN, F. CHERIAUX, R. DRELON, J. P. GAUSSOT, B. MARION, S. MAURIN, D. PICHOT, G. SANCERNI, G. VOISIN, and P. LEGAUD
Electricite de France
Research and Development Division
6, quai Watier
78400 Chatou, France

ABSTRACT

3SE is an expert system which monitors electric power supplies in a 900 MW PWR nuclear power plant. It is mainly used :

- as a continuous and real time aid to processing electric-equipment-induced failures,
- as an aid to preparing the electric equipment removal from service,
- as an aid to training operators in the use of electric power supplies,
- for managing the technical documents on electric power supplies.

This system largely relies on the techniques of Artificial Intelligence and data bases. It is based on the thorough knowledge of the facility topology and operation as well on principles of qualitative physics.

At present, we have reached the final phase of the program assembly, and no major problem remains. The system will be definitely completed by the beginning of March 1989 and it will be implemented in Bugey unit 2 by June 1989. The research effort required was 16 engineer - years. It is a major breakthrough in artificial intelligence and operator aid.

INTRODUCTION

The penetration of conventional data processing in the field of nuclear power plant monitoring and operation is difficult because of the multiplicity of operating aspects that have to be taken into account and which are not easily programmed by conventional means.

Artificial intelligence technology may well provide a solution to this problem. Indeed, it uses new techniques to manage data bases and makes it possible to formalize the operators' expertise and to automate the management of operators' technical knowledge [2].

Studies on expert systems applied to operator aids began in 1984.

In 1986, the merits of expert systems in this field were demonstrated with a demonstration prototype coupled to an operator training simulator [1].

Simultaneously, the need for closer monitoring of electric power supplies in PWR power plants was felt. Indeed, incidents of the power supply loss-type are characherized by :

- tricky diagnosis,
- impact on safety-related systems and controls,
- significant occurrence probability.

It was therefore decided to continue the development effort in an industrial environment and, to this end, to install a system in one of the PWR units of the Bugey nuclear power plant.

AIMS

This monitoring system is a computerized aid designed to help the operator process all the electrically-induced failures occurring in the facility.

It is used for following purposes :

- permanent and real-time processing of events caused by electric power system failures. This function is provided by an expert system giving a detailed picture of the facility state, thus enabling a diagnosis to be made.

- preparation for the whithdrawal from service of electric equipment. For this function, an equipment data base and a simulator are used.

- operators' training in the use of the unit power supplies. This function is provided by an equipment data base, a simulator and an explanation facility associated whith the diagnosis-aid expert system. Thanks to this explanation facility, the operator knows how the system reached the diagnosis.

- management of the technical documents on electric power supplies. With this function, reports are automatically edited from the equipment data base.

METHODOLOGY

This application uses the techniques of Artificial Intelligence and data bases to a large extent. The system is based on a two-level structure reflecting the separation between the design aspect and the on-line application one.

Design

The design part of the system fulfils the following functions :

* acquisition, management and checking of the basic system data and knowledge,

* formatting of these data for efficient real-time processing.

The architecture of the design system is illustrated in figure 1.

At the center of the system, there is a single data base providing a detailed description of the equipment and operation of the plant electric power supplies [5].

This set of data consists of :

- topological data describing the component nature and their connections,
- data describing normal equipment configurations according to the state of other components or of the overall plant conditions,
- functional data describing the behavior of components in case of proper operation or malfunctions as well as the conditions in which given operating procedures must be followed.

A number of treatments are associated whith this data base; i.e :

* a full-screen dialog for data acquisition. At present there are 20 different data acquisition grids, each corresponding to a type of equipment. An example of data acquisition grid can be seen in figure 3.

* checks (some 400) subdivided into :
  - syntactical checks made during data input,
  - overall checks to make sure the data base is complete,
  - semantic checks to make sure the data base is functionally correct,

* generation of nomenclature-type files for the real-time system,

* editing of technical reports on the components,

* "expert system" processing specially adapted to complex data handling in order to generate GENESIA I rulesets performing the identification, diagnosis and simulation functions in real-time.

These treatments are made by the GENESIA II inference engine. The facts base consists of two parts :

- one part containing the detailed facility description,

- one part specifically corresponding to the functions to be generated (Identification, Diagnosis, Simulation). It contains a model of the component operating principles.

The ruleset automatically writes GENESIA I rules. It is based on a set theoretic logic explicitly intregrating quantifiers.

It should be noted that the data base describing a given facility, as defined above, is not restricted to a particular application but covers a whole range of applications of the operator-aid type (such as those described in this paper) and of the computer-aided reliability study type...

Similarly, the various facts bases describing a given application are adapted to all the facilities described with data bases having the same structure.

The expert system shell chosen at this level is the GENESIA II. It consists of an inference engine based on predicate calculus and using forward chaining.

The various processing modules actually performing the monitoring system functions are automatically generated from the same data. Thanks to this approach, the homogeneity and consistency of each module in the system is guaranteed.

Moreover, with this architecture, the system is readily adaptable when the plant is modified or when functions are extended.

On-Line Application

The on-line system architecture is shown in figure 2.

Each of the monitoring system modules is a specialized expert system performing one of the required functions.

- real-time identification of the state of the plant electric power supplies,

- real-time diagnosis and choice of the procedure with, upon demand, the reasoning which led to the diagnosis,

- simulation of the electric power supplies behavior based on an actual plant state (prediction) or a reference state (training).

In fact, these modules are rulesets pertaining to a given application and plant.

The facts used by these rules are on/off or analog data automatically acquired on the unit computer and data processing system.

The expert system tool chosen at this level is tailor-made from the GENESIA I real-time inference engine. For performance and portability's sake, this engine is written in C language. It is based on propositional calculus and uses forward chaining.

This two-level structure combines the short execution time of the GENESIA I R.T. inference engine for on-line processing and the GENESIA II language capability for the design of large knowledge bases.

SCOPE

The monitoring system monitors all the electric power supplies of a 900 MW PWR nuclear power unit excluding the thermal-hydraulic phenomena.

To be more precise, the field of application encompasses :

- all the components, together with their controls, supplying electric power to the unit and shared systems (3000),

- the actuators of the pumps and valves controlled from the control-room, together with their controls (250),

- The process instrumentation channels,

- the alarms (2000 indicating lights) in the control room as well as the unit computer system variables (5024 on/off values and 1550 analog values),

- the reactor protection channels,

- the control room recorders and indicators (250).

This amounts to over 11 000 components described by 150 000 basic data.

FUNCTIONS DESCRIPTION

The application centers around the following four main functions :

- plant status identification,
- diagnosis,
- simulation,
- consulting an equipment data base.

Real-Time Identification of the Plant State

This function is aimed at producing an as detailed as possible reconstruction of the state of the plant electric power supplies only using the data provided by the unit computer.

This same reconstruction process is also used for the diagnosis and therefore takes place on a permanent basis. The operator merely consults the results.

Through this process, the state and availability of each component involved in the application (pump, valve, actuator, alarm, protection channel ...) are identified.

This function is performed by a specialized expert system. In a first phase, this system translates the values of the data acquired by the unit computer into component states.

In a second phase, using its knowledge of the plant topology and operation, the system completes step by step and simultaneously validates the partial plant state obtained during the first phase.

This approach has several advantages. Among others, it has the merit of :

- separating the description of the information sent by the instrumentation from the description of the topology and operation of the facility itself, thus facilitating the system updatings,

- providing a valuable aid for processing instrumentation failures by spotting inconsistencies in the data from the unit computer [3].

Diagnosis

The diagnosis function is provided to informe -in real time- the operator of any abnormal event occuring in the facility. The diagnosis is made by a special expert system which interprets the result of the unit state identification function. This expert system is based on cause-consequence relationships between components and on the concept of normal conditions.

In practice, the processing can be divided in two phases :

- based on the alarms recalculated from the data sent by the unit computer system, the system draws up a list of potential failures,

- based on this list and on the identified state of the facility, the system selects the only potential failures which are relevant considering the unit state and which do not result (even indirectly) from another identified failure.

The diagnosis consists of :

- a synthetic piece of information describing the overall state of the plant : the name of a procedure, or of a typical condition for instance,

- existing or relvant equipement failures pointing out to the operator the areas on which to concentrate his attention to restore more normal operating conditions.

This processing is continous. Note that in the chosen approach, there is no restriction on the number of failures which can occur simultaneously.

Simulation

The behaviour of the plant power supplies can be simulated starting from :

- a reference state,

- a state directly stored from the information provided by the unit computer system,

- a state previously derived by simulation.

The operator modifies initial state as he deems necessary and then starts the simulation. As a result, a so-called final state is derived, which corresponds to the equilibrium state reached by the plant at the end of the transient.

From a practical point of view, this simulation is run by an expert system based on the knowledge of the facility topology and operation. The overall simulation is achieved by combining the component local behaviors.

Consulting the Equipment Data Base

This function enables the operator to search a data base for the static information concerning all the components involved in the application.

MAN-MACHINE INTERFACE

The monitoring system will be moderately used under normal conditions by very different users with no particular training in the computer technique :

- the operating crew members in the control room,

- the maintenance personnel.

These facts as well as the system location in the control room, among numerous other data and operating means, has led us to choose a straight-forward man-machine interface largely resorting to the concept of window and to mice.

As far as the hardware is concerned, this interface consists of two totally independent and interchangeable workstations. One is located in the control room and the other in a separate room used by the power plant maintenance personnel. A station is made up of :

- a color graphics display unit,

- a mouse with a click button,

- an alphanumeric keyboard.

To dialog, the main device available is the mouse with its button. The use of the keyboard has been on purpose reduced to a minimum.

The image seen by the operator on the screen at a given instant is a combination of a number of basic graphic objects.

There are four basic graphic objects :

- the icon, which is a simple and compact means of access and progression in the dialog,

- the dynamic process diagram, which is a graphical animated representation of a part of the electric power supplies,

- the list, which contains a set of components and data on the plant state that can be consulted, printed or pointed at,

- the data acquisition grid, which is a special list used to feed alphanumeric data into the computer via the keyboard.

HARDWARE AND SOFTWARE CONFIGURATION

As regards the hardware, two BULL SPS7 computers are used for the application :

- a so-called application computer connected to the unit computer and data processing system through an ETHERNET like network and which performs all the monitoring system functions,

- a so-called design computer supporting the knowledge-base describing the plant.

It is used in particular :

. to update the data and knowledge base whenever the plant is modified,

. from this data and knowledge base , to automatically generate specialized expert systems pertaining to the various monitoring system functions.

This architecture has the merit of minimizing the monitoring system unavailability during plant modifications as well as the costs if the system is to be adopted in the 4 Bugey PWR units. Indeed, in this case, there will be 4 application computers and only one design computer which will be devoted to the management of the data and knowledge bases of all 4 units.

The application computer incorporates, among others :

- three processing modules, using a 32-bits microprocessor (68020), running in parallel under the SPART Operating System, a 26 Mbyte memory and two 150 Mbyte disks,

- two color graphics workstations with a mouse each.

Software used :

- C language,
- GDS graphic software,
- ORACLE data base management system,
- CALLIOPE (PDL) spécifications.

The design computer is made of a single 32-bits microprocessor (68020) module run under UNIX, a 12 Mbyte memory and two 150 Mbyte disks.

Software used :

- C language,
- ORACLE data base management system,
- GENESIA II expert system,
- CALLIOPE (PDL) specifications.

RATE OF USE

Beside its self service use for training purposes the system will also be used :

- approximately twice a month for minor incidents,

- approximately once a year for a more severe failure,

- during plant shutdown when maintenance operations can be carried out on several busbars at the same time.

EXPECTED RESPONSE TIMES

Data Generation

The data are generated in two steps :

- checks : 3 hours,
- file generation for real-time processing : 3 hours

These response times are perfectly compatible with the rate at which the system is updated, i. e, one update every week or every other week.

Real-Time Processing

The diagnosis is reached in 7.5 seconds so that 15 seconds at most elapse between an alarm and the diagnosis.

The maximum response time the equipment data base is consulted is 20 seconds.

DEVELOPMENT TEAM AND SCHEDULE

The members of the team working on this projet come from :

- the Bugey power plant : these are responsible for staking the needs, collecting the data, analyzing the operation,

- the Research and Development Division : these are in charge of industrial data processing aspects, man machine interface, processing of the data bases and expert systems, operation analysis and data structuring.

Furthermore

- the SEMA-METRA company developed the link between the monitoring system and the unit computer system.

- the STERIA company wrote a version of GENESIA I in C language.

This work amounts to a 16 man-years effort, Electricite de France's share representing 80 % of this figure.

The development schedule is :

JANUARY 1987	:	BEGINNING OF SPECIFICATIONS DRAFTING
MARCH 1987	:	SPECIFICATIONS COMPLETED
JUNE 1987	:	TECHNICAL AND COMMERCIAL EVALUATION
NOVEMBER 1987	:	BEGINNING OF DEVELOPMENT STUDY
SEPTEMBER 1988	:	PROGRAM ASSEMBLY TESTS
MARCH 1989	:	ACCEPTANCE TESTS AT THE FACTORY
JUNE 1989	:	ACCEPTANCE TESTS AT THE BUGEY POWER PLANT
SEPTEMBER 1989	:	VALIDATION (DECENNAL INSPECTION).

CONCLUSIONS

At present we have reached the final phase of the program assembly, and no major problem remains.

The following main conclusions can be drawn from our experience :

- data bases or expert system shells are flexible tools,
- a mass of data have to be collected, entered, checked and updated, and their quality must be verified,
- the system may be applied to large-size facilities thanks to a methodology based on the acquisition of thorough knowledge [4].

The development of this project has enabled us to acquire a unique experience in the management and implementation of such industrial-size, systems. Moreover, we will thus be able to preserve our advance in this field by taking into account operating experience.

Finally, this system represents a step forward in the long term development of data bases and knowledge bases on PWR units. Thus, studies will be automated to a large extent from the design stage (reliability, operating rules ...) to the operation stage (aids to operation and monitoring...).

REFERENCES

1. ANCELIN J. - GAUSSOT J.P - LEGAUD P.. EXTRA . a Real Time Knowledge - Based Alarm System. A.N.S. Meeting - "A.I. and other Innovative Computer Applications in the Nuclear Industry" Snowbird - 1987.

2. WATERMAN D.A.. A Guide to Expert System. Addisson - Wesley Publishing Compagny - 1986.

3. OSBONE R.L. - GONZALEZ A.J. - BELLOWS J.C. - CHESS J.D.. On Line Diagnosis of Instrumentation Through Artificial Intelligence - 1985 Westinghouse Electric Corporation Power Generation Operations Division Orlando, FL. 32817.

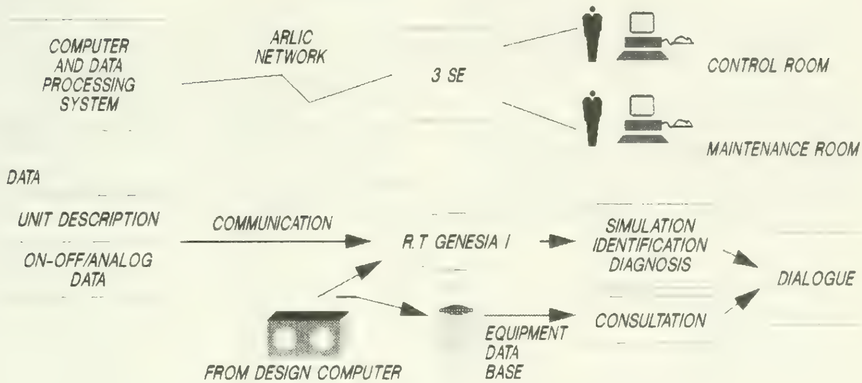
4. DORMOY J.L.. Methodologies pour des systèmes experts de seconde génération - 1988 - A paraître

5. HAYES - ROTH.. A Blackboard Architecture for Control. Artificial Intelligence. vol 26 pp 252-321 July 1985.

EDF DIRECTION
DES ETUDES ET RECHERCHES

SYSTEM ORGANIZATION

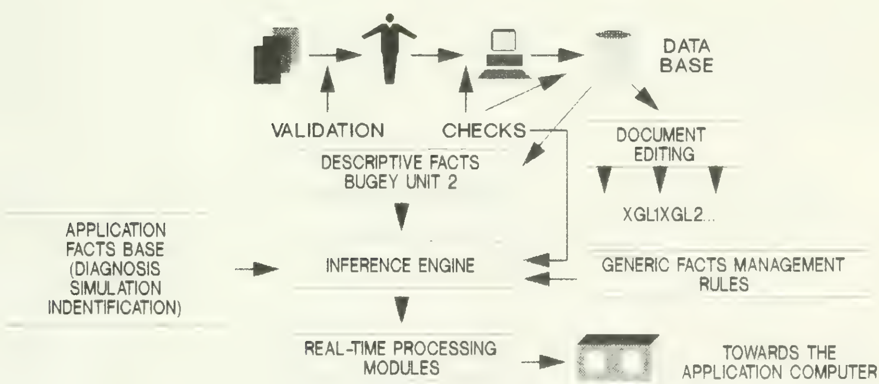
APPLICATION COMPUTER



EDF DIRECTION
DES ETUDES ET RECHERCHES

SYSTEM ORGANIZATION

DESIGN COMPUTER



EDF DIRECTION DES ETUDES ET RECHERCHES

COMPUTER AND PROCESSING SYSTEM DATA INPUT FORM

3 SE PROJECT UNIT COMPUTER SYSTEM VARIABLE CONSULTATION

ID.NB : 2LNA506EC...

TYPE : ON/OFF (ON-OFF/ANALOG)

COMPUTER SYS. MIN. CB. : 2 KIT 027UP...

INHIBITING VARIABLE : 2LNA551EC...

ID. NB OF UPSTREAM CONTROL MODULES :

DESCRIPTION : 2LNA IS USED ON MAINS

MODE : N... (N. AST. AT)

COMP. SYST. PCB : 65

MIN.CD.OF RELAY SYST.TO COMPUTER SYST. :

2LNA 001UP

MONITORED EQUIPMENT	ATTRIBUTE	ATTRIBUTE VALUE	VALUE
: 2LNA 001CS...	: SUPPLY...	BACK UP...	: 1
: 2LNA 001 TR...	: VOLTAGE...	PRESENCE..	: 1
: 2LNA 001 CS...	: SUPPLY...	NORMAL...	: 0
: ...	: ...	: ...	: ...
: ...	: ...	: ...	: ...

Alarm Processing System

P. DI DOMENICO and E. MAH
Bechtel National, Inc.
50 Beale St.
San Francisco, California 94105, USA

J. K. CHAN
Bechtel Power Corp.
50 Beale St.
San Francisco, California 94105, USA

D. CORSBERG
INEL, EG&G, Inc.
P.O. Box 1625
Idaho Falls, Idaho 83415, USA

J. NASER
Electric Power Research Institute
3412 Hillview Avenue
Palo Alto, California 94304, USA

J. SOMSEL
Pacific Gas and Electric Co.
77 Beale St.
San Francisco, California 94016, USA

E. SCARL
Boeing Computer Services
P.O. Box 24346
Seattle, Washington 98124-0346, USA

ABSTRACT

During power plant transients, a control room operator is deluged with large amounts of diverse information of varying degress of importance and in different formats. Control room operators are sometimes overwhelmed. They are required to make quick decisions based on operating procedures thereby bypassing most deep-level reasoning. The application of expert systems to process alarms and to perform diagnosis can substantially improve the quality of information presented to operators.

The high-level goal of the Alarm Processing System is for operators to enthusiastically accept a new technology that will improve their response to alarms during plant transient conditions. The Alarm Processing System dynamically prioritizes alarms based on the state of the plant; reduces the amount of information presented to the operator by grouping and displaying alarms in accordance with the present state of the plant; and allows nuisance alarms to be suppressed.

Object-oriented programming techniques are used to describe plant analog and binary sensors, alarms, and plant states. The system reasons about nuisance signals taken off-line by using a logic system that has three states: true, false, and null. A null or invalid state is propagated through the logic until the operator can assign a derived value for the signal.

INTRODUCTION

A typical nuclear power plant has approximately 2,000 alarms and displays designed to be useful aids to the operators. During plant transients, mode changes, and equipment trips, hundreds of alarms may be activated in a short period. This data bandwidth is more than the human nervous system can effectively handle. Many of these alarms do not contribute new information to assist the operator's diagnosis. Alarm information has varying degrees of importance and is presented in multiple formats. The volume and diversity of information are overwhelming and, consequently, the operators are forced to use "tunnel vision" procedures and step through a rigorous sequence of plant status verifications. Based on industry experience, they follow step-by-step procedures that bypass much of the deep-level reasoning that is needed if each process indication is treated with equal importance. Thus, finding important information and arriving at the root cause of the event is delayed. Their overall performance naturally degrades during these high-stress periods.[1] The application of expert systems to process alarms and perform diagnosis can substantially improve the quality of information presented to the operator.

Electric Power Research Institute (EPRI) is funding the development of an Alarm Processing System (APS) which will be demonstrated on Pacific Gas & Electric Co.'s Diablo Canyon Power Plant training simulator. Pacific Gas & Electric Co. is supporting the project by making available skilled operators and other domain experts as well as time on its simulator.

PROJECT SYSTEM GOALS

The highest goal of the project is to develop and demonstrate a system that will improve the response of nuclear power plant control room operators to alarms. The system must be enthusiastically accepted by plant operators to be considered a success. The system will utilize expert system technologies, as required, to reason about alarms.

The system design is based in part on a series of interviews with Diablo Canyon Power Plant operators that focused on the shortcomings of the current alarm systems and operators' perceived needs. The system has been designed to fulfill an operator's expectations expressed as follows:

- Alarms must be dynamically prioritized based on the state of the plant.
- Alarm prioritization information must be presented without operator action.

- The system must help the operator quickly identify serious problems that may be outside of his focus during a transient or plant trip while he is working on Emergency Operating Procedures (EOP).

- The system must help monitor "background" conditions which the operators deem important and alleviate nuisance alarms while not suppressing information.

DISPLAY CONCEPTS

Figures 1 through 4 show a proposed user interface concept for the APS. We are working with operators to determine appropriate displays.

Figure 1 illustrates the display screen areas of the APS. The screen is divided into four functional areas: the state display area, the critical events display area, the additional information display area, and the command line area.

The state display area displays a rectangular region on the screen for each of the plant states. Active plant states are highlighted. The name of the plant state is displayed with each of the regions. An alternative method for displaying these states might be to display only the active states. The states display area is never masked by other displays and is always visible.

The critical events display area displays high priority events. Events are displayed in a chronological order with the latest events occupying the bottom of the list. The APS displays events in this list when it wishes to emphasize those events. These events may be alarms or other important messages. The critical events display area is never masked by other displays and is always visible.

The additional information display area provides an area for displaying more information on any of the objects in the system. Figure 2 shows how the APS displays additional detailed information on states. As shown in Figure 3, detailed information on high priority events may also be displayed in this area.

As shown in Figure 4, the command line area provides a mechanism for additional operator requests for information. The options provided here might include requests for information not available through the other display areas.

Display design will be integrated with the Diablo Canyon Power Plant control room man-machine interface guidelines as to consistent color coding, viewing distances, etc.

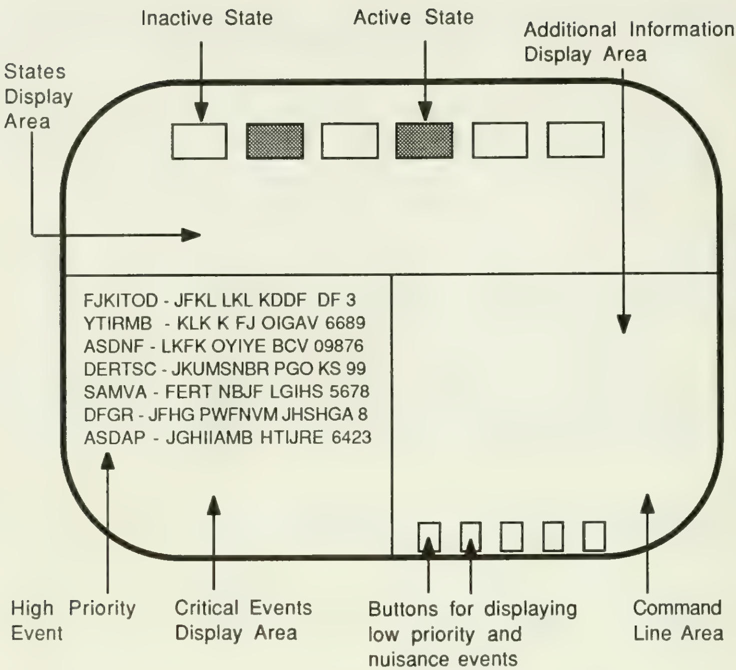


Figure 1 APS Normal Display on the CRT

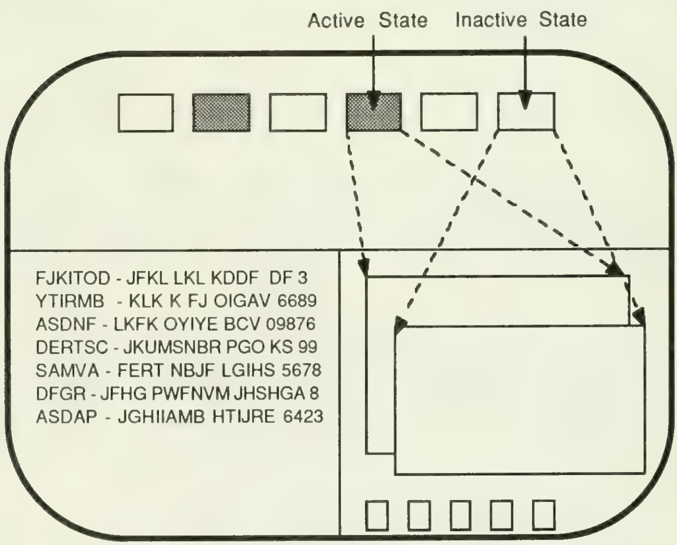


Figure 2 APS Detailed Information on States

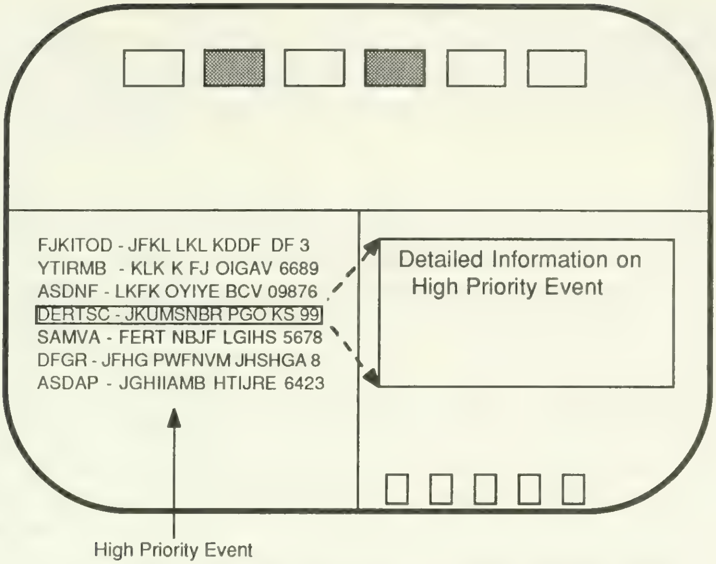


Figure 3 APS Detailed Information on High Priority Events

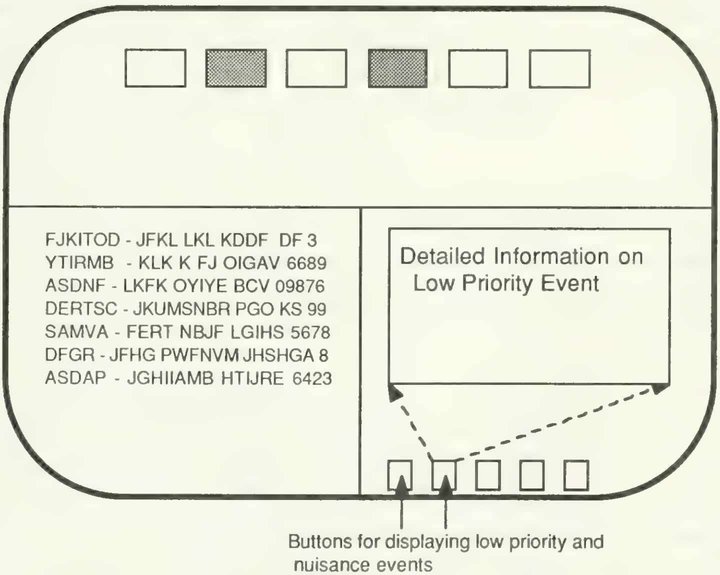


Figure 4 APS Detailed Information on Low Priority Events

CENTRAL CONCEPTS OF APS

An important feature of APS is to focus the operators attention on critical or important pieces of information. It does so by summarizing the alarm information presented to the operator. Plant states and events cause alarms to be displayed in groups and priorities as described below.

Plant States

A plant state represents some well-defined situation or operating mode in the plant. The APS recognizes plant states by evaluating a logical combination of plant parameters based on plant binary and analog parameters. The values of these plant parameters are obtained though plant data acquisition systems.

Emergency Operating Procedures (EOPs) are a good knowledge source for defining a subset of the plant states. The APS identifies plant states as active, based on plant inputs defined by entry conditions of EOPs as well as other well-defined plant conditions. The APS, however, does not perform EOP tracking. Tracking the EOPs would require operator input and place additional burdens on the operator. States are used for prioritization and grouping.

Logical expressions are used in the APS as a way of capturing the knowledge for a plant state activation. Plant states and other plant conditions are identified in the system as active when an expression representing the entry condition for that state is satisfied. This expression consists of a well-defined boolean operator on a list of symbols representing parameters in the system.

Coalescing

The APS displays the active plant states on a CRT. The APS uses active plant states to reduce the alarm information presented to the operator when alarms present no more information than the active states. This feature embodies the concept of coalescing alarms. Alarms that are coalesced are maintained in a list within the plant state. Alarms that are the cause of a plant state may be considered for inclusion under coalescence. Furthermore, alarms that are expected to be direct consequences of these states may also be coalesced. An active alarm that is coalesced is removed from lists that display critical information to the operator. The alarms that remain uncoalesced are those that reflect anomalous situations that are not explained by a plant state, either because they are not affected by that state (independent alarms) or because they are not what was predicted for that state (expected alarms).

Expected Alarms

The APS emphasizes alarms that are expected during a plant state when those alarms have not occurred. The APS includes alarms that are expected in the coalesced alarm list. If these alarms are not received within a given time limit, the system emphasizes them as alarms that have not occurred.

Internal Alarms

Certain plant conditions are considered important only during specific situations. These conditions are specified by predefined thresholds of plant parameters and are called internal alarms by the APS. Examples of these are the "fold-out" conditions that are part of the EOPs. The APS monitors and emphasizes those conditions only during the time that the situation represented by the plant state is active. Plant states maintain a list of internal alarms that correspond to those predefined thresholds. Some of these important plant parameter thresholds may not be alarmed by the existing alarm system. When the plant state is activated, it enables its internal alarms. When the predefined thresholds are met, the APS then provides some indicator to emphasize this event. Logical expressions are also used to capture the knowledge of when an internal alarm is active.

As an option, the APS allows operators to request the monitoring and emphasis of these internal alarms even when the associated states are inactive.

Prioritization

An alarm may be raised or lowered in priority by an active state. In the context of an alarm that might also be coalesced, raising the priority of that alarm will supercede any attempt to coalesce it. For the APS, raising the priority of an alarm means that that alarm is sufficiently important in the current context that it is always displayed when it is activated. Conversely, lowering the priority of an alarm means that that alarm has little significance when a particular plant state is active.

Alarms may also be lowered in priority based upon their relationships to other events/alarms. Corsberg defines these types of relationships in his work with the Alarm Filtering System (AFS).[2] A "Direct Precursor" relationship between alarms implies a causal relationship between them. The alarm associated with an event that is a direct consequence of another event may be lowered in priority on this basis. A "Level Precursor" functional relationship between alarms exists when there are multiple alarms with different setpoints on a parameter. In this

case, when more than one alarm is active, only one of the alarms is important and the others may be lowered in priority.

The APS allows an operator to de-emphasize alarms that are related to failed instrumentation and systems that are out of service for maintenance and testing. When an alarm is considered out of service or a nuisance, plant personnel will be allowed to mark this alarm as nuisance priority. Prior review and approval of relationships could allow APS to suppress alarms or auto-acknowledge alarms. This could allow reduction in distractions to operators.

There are three levels of priority: high, low, and nuisance as defined below:

- High Priority. Events that are high priority are important for the operator to know about. These events are important to the safe operation of the plant or indicative of destructive conditions. These events may also be the causal events in a sequence of events. Consequential events may also be high priority if they are important for other reasons such as safety or criticality.

- Low Priority. Events that are low priority are those that do not require immediate operator response and are not as important for the operator to see. These events may be consequential rather than causal.

- Nuisance Priority. Events that are nuisance priority are those in which the information is inappropriate for the current state of the plant or subsystem or information which is wrong because of out-of-service equipment or sensors.

SCENARIO

The following plant transient scenario demonstrates some APS concepts. Our nuclear power plant consists of two units. Each unit is a four-loop pressurized water reactor. At the onset of the transient, Unit 2 is operating at full power whereas Unit 1 is operating at 75 percent power. At 09:32:52 Unit 1 main feedwater pump 1-2 is tripped by high vibration. The operators respond by ramping down Unit 1 to accommodate the loss of the main feedwater pump. At 09:41:41, Unit 1 experiences a reactor trip. The following is a list of alarms leading up to and following the reactor trip (without the APS interpretation):

09:30:22	R0425	AUX SALT WATER PUMPS ROOM DOOR OPEN
09:32:42	R0573	MFW PUMP TURB 1-1 HI VIBRATION
09:35:52	R0557	MFW PUMP TURB 1-1 TRIP
	R0566	MFW PUMP TURB 1-1 HP BRG OIL PRESS LO
09:36:09	R0343	STM GEN 1-2 LEVEL LO FROM REF
	R0346	STM GEN 1-3 LEVEL LO FROM REF
	R0348	STM GEN 1-4 LEVEL LO FROM REF
	R0369	STM GEN 1-1 LEVEL LO FROM REF

09:37:29	R0575	MFW PUMP TURB 1-1 ZERO SPEED
09:41:41	R0070	STM GEN 1-4 LEVEL LO 1/2
	R0241	TAVE DEVIATION TAVE-TREF HI
	R0176	STM GEN 1-4 LEVEL LO-LO 1/3
09:41:41	R0027	STM GEN 1-4 LO-LO LEVEL 2/3 REACT TRIP
09:41:41	R0492	UNIT 1 REACT TRIP
	R0473	ROD CONTROL SYS URGENT FAILURE
	R0478	ROD POS INDICATION ROD BOTTOM
	R0551	ROD POS INDICATION RODS AT BOTTOM
	R0713	EH SYS PRESSURE LO
	R0703	TURB AUTO STOP OIL SOLENOID TRIP
	R0089	TURB AUTO STOP OIL PRESSURE LO 1/3
	R0049	TURB STM STOP VLVS CLOSED 1/4
	R0010	TURB TRIP AND P7 2/2 REACT TRIP
	R0331	SOURCE RNGE NC31 LOSS OF DET V
	R0332	SOURCE RNGE NC32 LOSS OF DET V
09:41:46	R0534	PZR PRESSURE LO CHAN 474
	R0535	PZR PRESSURE LO CHAN 457
	R1038	TURB EXH HOOD-C SPRAY VLV OPEN
	R1037	TURB EXH HOOD-B SPRAY VLV OPEN
	R0038	LO TAVE 2/4
	R0864	TURB EXH HOOD-A SPARY VLV OPEN
	R0080	FW ISOL FROM REACT TRIP AND LO TAVE 2/4
	R0080	LO TAVE 1/4
	R0069	STM GEN 1-3 LO LEVEL 1/2
	R0067	STM GEN 1-1 LO LEVEL 1/2
	R0068	STM GEN 1-2 LO LEVEL 1/2
	R0171	STM GEN 1-2 LEVEL LO-LO 1/3
	R0164	STM GEN 1-1 LEVEL LO-LO 1/3
	R0073	STM GEN 1-3 LEVEL LO-LO 1/3
	R0024	STM GEN 1-1 LO-LO LEVEL 2/3 REACT TRIP
	R0025	STM GEN 1-2 LO-LO LEVEL 2/3 REACT TRIP
09:41:53	R0026	STM GEN 1-3 LO-LO LEVEL 2/3 REACT TRIP

At 09:32:42, a high-vibration alarm is received in one of the two main feedwater pumps, which subsequently leads to the tripping of main feedwater pump 1-1. The tripped main feedwater pump causes a LOW PRESSURE CONDITION IN THE HIGH PRESSURE BEARING OIL (R0566) and ZERO SPEED DETECTED IN THE PUMP TURBINE (R0575). Since these two alarms are the direct result of tripping the pump, they can be given a lower priority by the APS and are removed from the critical alarm list. Therefore, at 09:37:29, our hypothetical APS CRT display may look like Figure 5.

In our example, the APS detects that ReactorTripped is an active plant state by evaluating the following expression:

(DECREASING NEUTRON FLUX and RODS ON BOTTOM and TRIP BREAKERS OPEN)

The values of parameters in this expression are themselves composed of expressions, and so forth, until the expression objects are plant binary inputs or relations with analog inputs.

All the alarms that come in after 09:41:41 are normally expected when the reactor is tripped. In the APS, these alarms are coalesced into the state of ReactorTripped and are not displayed individually. Figure 6 shows the APS CRT screen after Unit 1 has been tripped. The 41 alarm messages which are conventionally presented are reduced to 4 messages by the APS.

The APS will warn the operator if an expected alarm (i.e., ROD POS INDICATION RODS AT BOTTOM [R0551]) is not received. This warning would appear on the critical event list. The operator may interpret this warning as one indication that the reactor did not scram.

An APS internal alarm would occur if, for example, the level in the condensate storage tank has dropped below 10 percent. This event indicates to the operator that he needs to switch to an alternate source of auxilliary feedwater.

In our example, the AUX SALT WATER PUMPS ROOM DOOR OPEN (R0425) alarm may still be active when the reactor is tripped. However, this alarm is a lower priority alarm under these circumstances and the APS will cause this alarm to be lowered in priority because of the activation of ReactorTripped. This is an example of a plant state lowering the priority of an alarm.

OBJECT ORIENTED PROGRAMMING

Object oriented programming techniques are useful when there are many instances of a class of real-world things. An object is merely an abstraction of these things. All those things which share the same characteristics, for example being members of a group, are instances of an object. When information is captured for one of these objects, it may apply to all of them. Rules of behavior that apply to one of these objects may apply to all of them. Most things in the real world are described naturally by their membership in a particular group. The things that belong to the power plant control room can be described in these terms. In particular, there are objects which can be grouped such as analog inputs and binary inputs. Defining a technique for dealing with one of these things can be extended to others of this group.[3]

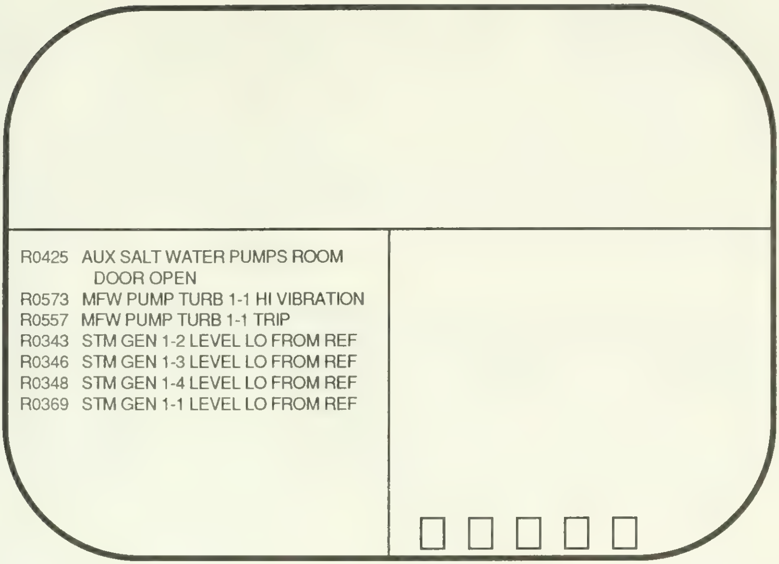


Figure 5 APS CRT Display before Reactor Trip

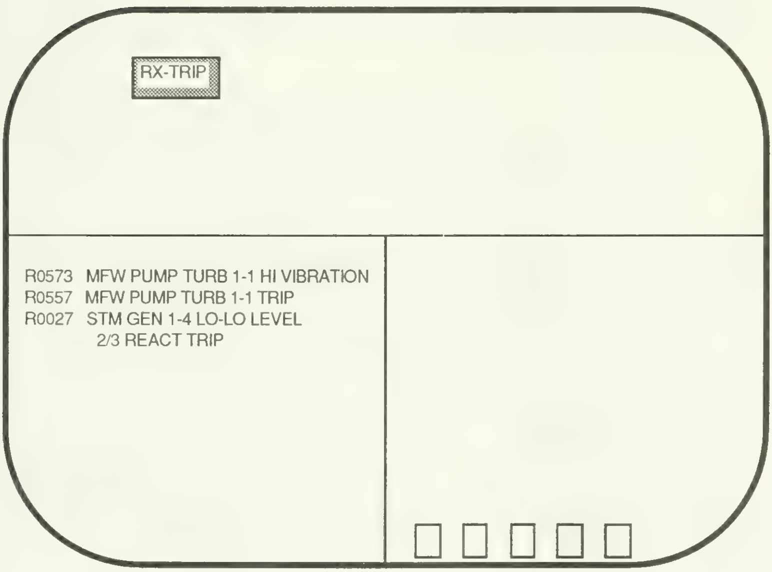


Figure 6 APS CRT Display Immediately After Reactor Trip

Object-oriented programming techniques enhance the modularity of capturing information about an object. Associating the information and behavior with an object is similar to what is done by human beings in the real world. This helps to bridge the gap between the computer code and real world objects.

There is no reason why an object can't also represent things that are purely abstract. Plant states are essentially abstract concepts that have meaning to control room operators. Categories of information that can be captured for one plant state might be useful to others.

The APS has been designed using object-oriented techniques. Object-oriented programming has its own terminology for describing a system. Slots are the attributes associated with an object for capturing information for that object. A method is a function associated with an object to perform some action on that object, often using the information captured in the slots of that object. A message is a call to an object that causes a method to be executed for that object. A message is said to be "sent" to that object.

APS object identifiers reflect their usage in the power plant domain. The objects used in the knowledge representation scheme for APS include the following:

- Analog Input Object. This object represents analog plant input data. Examples of such input data are pressurizer pressure and steam generator level. The value of the object is the analog value as displayed in the control room.

- Binary Input Object (alarm). This object represents binary plant input data. This object can represent a single binary input directly from the plant instrumentation. The identifier of the object is the tag number of the measuring device. The value of the object is the on or off value of the input signal.

- Alarm Object. This object represents the annunciator alarms and are usually a combination of binary inputs. The value of this object is either on or off.

- Plant State Object. This object represents the status of the plant or plant subsystem. The value of this object is either true or false, representing whether or not the state is active.

- Internal Alarm Object. This object is generated by the APS internally to monitor some particular events that are important when a plant state is activated. This object is enabled by the activation of a state which has this object as one of its internal alarms and is disabled by the same state deactivation. This object is active or inactive depending upon whether the event being monitored has occurred.

REASONING ABOUT ANALOG DATA

In order to reason about analog inputs, a logic object was invented to capture the value of a relational expression. A relational expression contains an operator such as "greater than," "less than," etc., that compares an analog parameter to another analog parameter or to a setpoint. Logic objects will have a value of true or false depending upon the evaluation of the relational expression. An example of a relational expression is:

(ANALOG-VALUE < 5)

If the ANALOG-VALUE parameter is 4 then the logic object has a value of T (true). If the ANALOG-VALUE parameter has a value of 6 then the logic object has a value of F (false). This object allows the system to reason about an analog input value with an object that has a boolean value. The APS is effectively translating an analog comparison to a binary value. This binary value is then used in the expressions to determine plant state activation.

REASONING ABOUT UNCERTAINTY

A problem faced by any diagnostic system is to recognize and then deal with data and instrumentation that is either suspect or known to be invalid. Questionable plant information can cause the operator's mental model to diverge from the physical process. Recognizing and remembering suspect or invalid data is a difficult step in itself. Knowing what the missing parameter means in terms of operational implications and emergency situations is even more difficult. Dealing with suspect data and the uncertainty associated with that data has long been a recognized need in computerized diagnostic aids. It is seldom recognized that operators themselves have extremely difficult time with this problem. Common practice in diagnostic aids is to present the operators with a conclusion and then a measure of that conclusion's validity or quality. This information quality measure is not particularly useful without an immediately clear and concise understanding of where that uncertainty is coming from.

The main purpose of the APS is to prioritize information that would be displayed in a plant transient. Taking alarms off-line can be accomplished through APS by a human, but other avenues for this will also exist. APS must, at the very least, be able to adequately deal with inputs that have been taken off-line or determined to be invalid. Within APS, combinations of signals and states are often used to represent plant conditions. Therefore, it is not enough to simply

place the object representing the input in an invalid state. The effects of that status must be propagated through any pertinent plant states. The meaning of a specific invalid data with all possible contexts cannot be anticipated. In reality, the current techniques for uncertainty management are inadequate to consistently deal with suspect data. Therefore, when the situation requires dealing with suspect information, it is prudent to call upon the operator to resolve the issue.

As mentioned earlier, APS utilizes combinations of inputs and other states to represent various plant states. These are represented by boolean expressions that reduce to values of either true or false. We are adding a third value called null. Null is neither true nor false and essentially delays final determination of the expression's value until it has been determined that the value is needed to provide information to the operator. For example, suppose we have the following states and inputs:

State1 = (InputA and State2 and InputB)

State2 = (InputC and InputD) or (InputF and InputG)

InputA InputB InputC InputD InputE InputF InputG

In addition, suppose that InputF is invalid. The effects of that status must be propagated into State2 and then into State1. However, we don't want to resolve the uncertainty unless that resolution is required for determining the value of State1 and/or State2. A simple algebra is used to not only propagate the uncertainty but to also delay its resolution until absolutely required. Each element in an expression is evaluated to a value of true (T), false (F), or null (N). If there are no null values, then the expression is treated in the same manner as a standard boolean expression. If there is a null value, it is combined as follows:

Input A and Input B

A \ B	T	F	N
T	T	F	N
F	F	F	F
N	N	F	N

Input A or Input B

A \ B	T	F	N
T	T	T	T
F	T	F	N
N	T	N	N

Returning to the example described above, InputF was invalid and therefore evaluated as a null. In examining State1, each element in State1's expression could be evaluated. Evaluating State2 would, in turn, cause its expression to be evaluated. Given the right combination of inputs to State2, that evaluation could return a null value (if InputC or InputD are false and InputG is true). Given the right combination of inputs to State1, its evaluation could return a null also. At this point, because State1 has importance in establishing priorities or in providing information to the operator, the null value must be resolved. That value is traced back through chain of expressions until the physical input is found. The operator is then informed that State1 could be valid (or invalid if that represents a change of value) depending on the currently suspect value of InputF. The operator is given the option of assigning a value or leaving the input in a non-conclusive state. Since APS is still evolving, further management of the invalid data is yet to be determined. A variety of options are being investigated including:

- Indicating any priorities or conclusions that are "tainted" by the invalid inputs.

- Periodically checking back with the operators on current status of previously requested data. The periodicity could be based on time (every shift change), operations (changing plant modes), or a combination of the two (every 2 hours if above 5 percent power). The periodicity could also be based upon the importance of the effects of the null value.

- Periodically checking back with operators on updated values if a value is left null. This could be handled in the same manner as above.

- Maintaining lists of all states that are affected by invalid or suspect data. These states could be viewed in the same manner as alarm points that are off-scan.

An important issue in dealing with invalid data is knowing when to "hand off" the problem. Care must be taken to not "hand off" the problem so often that the system becomes a burden to the operators (resulting in the system being ignored). Thus, many issues concerning the management of this aspect of the diagnostic aid remain.

MODEL-BASED REASONING

An alternative approach to assigning a null value to uncertain inputs would be to use model-based reasoning to analytically identify failed sensors and to calculate their "correct" values. Model-based reasoning offers a structured alternative for diagnosing faulty sensors and can operate when sensor data is missing or determined to be invalid.

The kind of model required is a behavioral simulation that can be used to monitor the system's operation. Assuming a particular set of system inputs for the model causes predictions to be computed for the system's sensors. The health of the system is determined by matching actual sensor readings against these prescribed values.

Once a model has been validated, any discrepancy between predicted and measured values indicates a failure in the physical system. Faults are located by generating hypotheses and testing those hypotheses against all available sensor data. Fault hypotheses are generated directly from sensor readings by inverting the functional relationships in the model.

A system called KATE which embodies these principles has been under development at the Kennedy Space Center since 1983.(4) A newer implementation of part of KATE, by Technology Applications, Inc. (Jacksonville, FL), is called ProSys and has supplied the object format for APS's knowledge base.(5) ProSys will be evaluated as a source of diagnostic inference to validate plant parameters and infer active states from them. Once the ability to validate data using ProSys has been established, it will be natural to use the validated parameters for state activation and the other purposes discussed in this paper.

EXPERT SYSTEMS TECHNOLOGY

The APS uses object-oriented programming and expression paradigm to capture knowledge. It can be described as an associative system in that it draws relationships or "associations" between the alarms and states. Rule-based systems are also associative, where the rules define the relationships between objects. In general, a rule-based system has an inference engine that performs the following scenario: if there is a change in the data, it will examine the antecedent of all of the rules and, if the antecedent is satisfied, perform the consequents of that rule. This will perhaps change other data in the system which may in turn repeat the cycle. This requires an extensive computing overhead which can cause rule-based systems to be slow. The APS does not use a rule base because of the overhead involved. In a sense, the use of expressions and messages capture some of the essence of rule-based systems. The expressions in our system behave like the antecedent of rules. The activation of states followed by the coalescing and prioritization behave like the consequents. In using objects, methods, and expressions to capture knowledge of the nuclear power plant more directly, we hope that we can provide a more timely response.

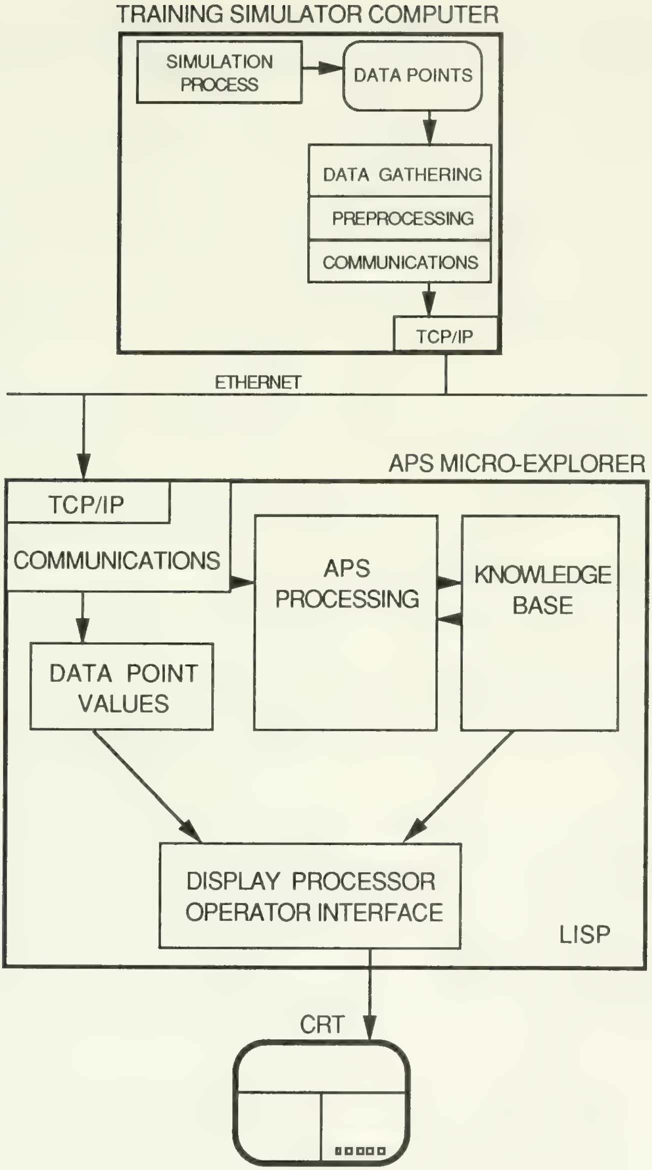


Figure 7 APS Architecture

SYSTEM ARCHITECTURE

The system design must incorporate the following functions:

- Obtaining Data. The data must be available to the APS. This function includes accessing data from the available sources which may include data that is archived from other computers or data that is available through a communications port.

- Communications. Communications mechanisms must exist between the different processors (physical hardware) and processes (software).

- Preprocessing. Preprocessing includes the following functions: handling changes of state for binary inputs including contact bounce and passing only changes up to the next level of processing; and handling changes of analog inputs and passing these changes along only when significant changes have occurred, i.e. when the value has changed by more than an absolute delta value (the delta value may also be a variable quantity that is set by the system).

- APS Processing. The APS performs the functions of alarm prioritization and coalescing. Signal validation and diagnosis may be performed.

- Operator Interface and Display of APS Processing. The operator interface must allow the operator to interact with the system to request additional information or to input data into the APS. The operator interface must display the current status of the plant. It may use icons or other objects displayed graphically or text to represent the status of the plant. Displays will be coordinated with control room man-machine interface standards.

- Health of System. The health of the communications link, preprocessor, and APS status must be checked to insure that the system is presenting valid information to the operator based on the input data, the processing of data, and the encoded knowledge base.

A diagram for the proposed architecture of APS is shown in Figure 7. The training simulator computer contains the simulation process and performs the data gathering, preprocessing, and some communications for the APS. A MicroExplorer performs some communications as well as APS processing, display processing, and operator interface functions.

FUTURE TRENDS

The APS is now being developed and will be demonstrated near the end of 1989 at the Diablo Canyon Power Plant training simulator. The system to be demonstrated will be a stand-alone CRT and printer which will advise the operator of the priority and grouping of plant simulator alarms. Specifications will be provided that describe both a stand-alone system and a system that can be embedded into

annunciator and plant computer systems. The representation of objects within the APS will be that of the EPRI-model-based expert system ProSys.[4] [5]

A potential enhancement of the APS would be to model plant components and systems in order to perform diagnoses. The next generation of systems may be software systems embedded into conventional plant systems that can reason about the status of plant components, systems, and failed sensors using real time signals from plant sensors. Pacific Gas & Electric Co. is making provisions in its current upgrading of the Diablo Canyon annunciator and plant process computer systems to support a future production-grade APS.

The underlying principles of the APS can be applied to other, similar problems. Characteristics of these problems are: 1) large, closely coupled datasets to be digested, 2) real-time or near real-time results required, and 3) priorities dependent upon system state. Examples include transmission grid monitoring and petrochemical plant process monitoring. In general, systems that have extensive System Control And Data Acquisition systems with fast operator diagnosis and response requirements are likely candidates.

REFERENCES

(1) "Power Plant Alarm Systems: A Survey and Recommended Approach for Evaluating Improvements," EPRI NP-4361, December, 1985

(2) D. Corsberg, L. Johnson, "A Nuclear Reactor Alarm Display System Utilizing AI Techniques for Alarm Filtering," Proceedings of the American Nuclear Society Topical Meeting on Artificial Intelligence and Other Innovative Computer Applications, Snowbird, UT, August, 1987

(3) S. Shlaer, S. J. Mellor, "Object-Oriented Systems Analysis Modeling the World in Data," Project Technology, Inc., Yourdon Press, 1988

(4) E. A. Scarl, "Diagnostic Tolerance for Missing Sensor Data," Proceedings of 1989 Goddard Conference on Space Applications of Artificial Intelligence, Goddard Space Flight Center, May, 1989

(5) R. A. Touchton, N.S. Subramanyan, J. Naser, "Model-based Reasoning Technology for the Power Industry," Proceedings of the Conference of Expert System Applications in the Electric Power Industry, Orlando, FL, June 5-8, 1989

IRIS: An Expert System to Aid Nuclear Operators

W. MALFARO and A. ZYGMONT
Department of Electrical Engineering
Villanova University
Villanova, Pennsylvania 19085, USA

ABSTRACT

This paper discusses development of ISOLATION RESET INFORMATION SYSTEM (IRIS), an expert system to aid nuclear plant operators during plant transients known as automatic containment isolations. IRIS is implemented using the Personal Consultant Plus expert system shell, taking advantage of the dBase III Plus interface. The design of IRIS is discussed as well as the system's current state of development. The use of expert systems for training operators is discussed. The importance of gaining regulatory acceptance of expert systems is presented. This issue will ultimately determine the extent of expert system use in nuclear applications.

INTRODUCTION

This paper discusses IRIS, an expert system currently being developed at Villanova University. It is designed to assist nuclear plant operators in diagnosis of and recovery from automatic containment isolations. The paper is divided into five main sections.

The first section discusses the Primary Containment and Reactor Vessel Isolation Control System (PCRVICS). The PCRVICS initiates closure of various automatic isolation valves if monitored system variables exceed pre-established limits. This action limits the loss of coolant and the release of radioactive materials from the reactor coolant pressure boundary, the primary containment and the reactor enclosure.(1) Operator-required response to automatic isolations include determining the cause, verifying that automatic action has occurred as designed and implementing the appropriate reset or bypass procedure. The operator is faced with a burdensome task during these transients due to the number of process parameters, automatic actions and required operator responses. IRIS will assist operators in assimilating available raw data thereby reducing uncertainty involved in operator decision by providing advice and rapid access to critical information. Use of IRIS in the control room offers the possibility of reducing operator error thus enhancing plant safety and reliability.

The second section discusses the development tools used in the construction of IRIS. The system is implemented using the expert system shell PERSONAL CONSULTANT PLUS (PC PLUS). A description of the inferencing abilities of PC PLUS and how they are applied to the design of IRIS follows. Both forward and backward chaining are used. PC PLUS's interface to dBASE III Plus is mentioned and considerations in choosing these tools are discussed.

The third section discusses the design and construction of IRIS. The overall structure of the expert system is described and an example consultation is examined to further illustrate the structure of the knowledge base. Although IRIS has been designed as an off-line system, conversion to on-line use is possible through application of PERSONAL CONSULTANT ONLINE. The benefits of this conversion are outlined. The method of encoding data into a knowledge base is discussed. The information contained in IRIS was extracted from general plant procedures, plant Technical Specifications and other existing documents. This information was then used to construct the knowledge base. A portion of this information was used to build a prototype which was successfully tested. The prototype is now being converted into a total system containing all the necessary information. The benefits of this design methodology are discussed.

The fourth section discusses use of expert systems as training aids. Operator training is enhanced through expert system use. This occurs because of increased exposure to organized information and because of the ability of some expert systems to explain how a solution was formulated instead of merely stating a conclusion. Additionally, engineers who develop expert systems gain a deeper understanding of the subject while encoding the information into an expert system knowledge base.(2) Use of expert systems during control room simulator training is discussed. This provides a way to introduce expert systems to utilities and regulatory personnel who will ultimately determine the extent of their use in nuclear applications.

The final section discusses the importance of gaining regulatory acceptance of expert systems. Other applications of expert systems to nuclear operations are mentioned. Discussion is centered around using expert systems to guide operators through complicated procedures and databases which they must now use to extract needed information.

A conclusion summarizes the development of IRIS and how its use can enhance personnel training, nuclear operations and overall plant reliability.

PROBLEM DOMAIN

The following section reviews the design of a typical BWR Primary Containment and Reactor Vessel Isolation Control System, identifies operator response to automatic system actuations, and describes how IRIS will aid operators in responding to these actuations.

<u>Primary Containment And Reactor Vessel Isolation Control System</u>

The PCRVICS includes the instrument channels, trip logics, and

actuation circuits that activate valve closing mechanisms to effect isolation of the primary containment or reactor vessel or both. The purpose of the system is to prevent the gross release of radioactive materials to the environment from the fuel or a break in the Reactor Coolant Pressure Boundary. The PCRVICS automatically isolates the appropriate process pipelines whenever monitored variables exceed preset limits. Pipelines that penetrate primary containment and communicate directly with the reactor vessel have two isolation valves: one inside primary containment and one outside primary containment. This arrangement is shown in Figure 1. To prevent a single failure from causing an automatic valve closure, instrument relay contacts in two channels must open which de-energize the initiation relay causing an automatic valve closure. Referring to Figure 1, the inboard valve will close if any sensor A input AND any sensor B input exceed their setpoints. The outboard valve will close if any sensor C input AND any sensor D input exceed their setpoints. The PCRVICS is made up of approximately 130 such valves and 115 such instrument channels monitoring 25 process parameters. The valves are separated into Groups based on the plant system which is affected. The process parameters are given alpha-numeric codes called Isolation Signals. Tables 1 and 2 specify these designations and their relationship to one another. As an example, referring to Tables 1 and 2, a Group IB isolation could be caused by either Isolation Signal B (Reactor Level 2 Low, Low) or Isolation Signal D (Main Steam Line High Radiation). This information coded as production rules comprises part of the expert system.

Operator Response To Automatic Actuations

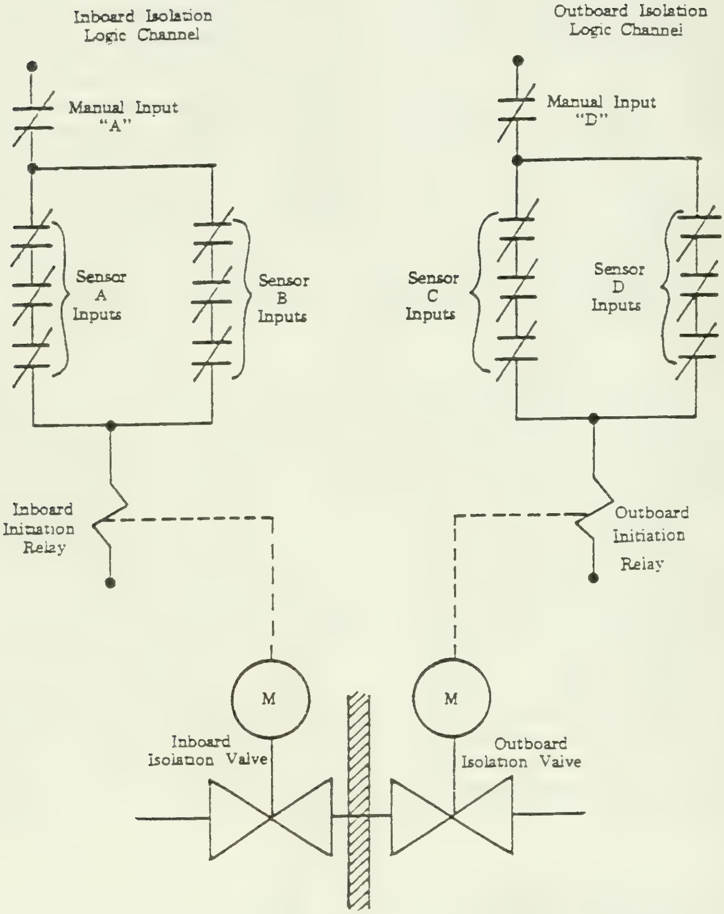
When an automatic initiation occurs, operators are to verify that all valves that should be closed are closed. Manual action must be taken if this has not occurred as designed. The operator must then, with the assistance of the technical staff, determine the cause of the automatic isolation. This involves determining what Isolation Signal caused the actuation, or more specifically what instruments caused the isolation. In general, once isolation is initiated, the valve continues to close even if the condition that caused isolation is restored to normal. The operator must manually reset the tripped logic and operate switches in the control room to reopen a valve that has been automatically closed. Unless a manual bypass under administrative control is provided, the operator cannot reopen the valve until the conditions that initiated isolation have cleared.

In summary, following an automatic isolation the operator must verify the isolation occurred as designed, determine what caused the isolation, and take manual action to bypass or reset the isolation as appropriate.

How IRIS Will Assist Operators

IRIS will assist operators to accurately analyze transients caused by the PCRVICS. From a human factors standpoint, operator errors can be classified into three broad categories: observation errors, analysis errors, and manipulation errors.(3) Use of IRIS offers the possibility of reducing analysis errors. This type of error occurs when the operator is aware and correctly knows the information

FIGURE 1



Inboard Isolation
Logic Channel
Manual Input
"A"
Sensor
A
Inputs
Sensor
B
Inputs
Inboard
Initiation
Relay
Outboard Isolation
Logic Channel
Manual Input
"D"
Sensor
C
Inputs
Sensor
D
Inputs
Outboard
Initiation
Relay
M
M
Inboard
Isolation Valve
Outboard
Isolation Valve

TABLE 1

GROUP ISOLATION	ISOLATION SIGNAL																								
	A	B	C	D	E	F	G	H	J	K	KA	L	LA	M	P	Q	R	S	T	U	V	W	Y	Z1	Z3
IA MSIV'S & STM LINE DRAIN			X	X	X	X									X	X									
IB MAIN STM. & REACTOR WATER SAMPLE LINES		X		X																					
IIA SHUTDOWN COOLING & HEAD SPRAY LINES	X																				X				
IIB RHR HEAT EXCHANGER SAMPLE LINES AND RHR DRAIN TO RADWASTE LINES	X							X																	
IIC RHR HEAT EXCHANGER VACUUM BREAKER LINES		X						X																	
III REACTOR WATER CLEANUP LINES		X							X														X		
IVA HPCI PROCESS LINES												X	X												
IVB HPCI TURBINE EXHAUST VACUUM BREAKER LINE								X					X												
VA RCIC PROCESS LINES										X	X														
VB RCIC TURBINE EXHAUST VACUUM BREAKER LINE								X			X														
VIA PRIMARY CONTAINMENT PURGE SUPPLY AND EXHAUST		X						X									X	X	X	X		X		X	X
VIB PRIMARY CONTAINMENT EXHAUST TO R.E E.C.E		X						X									X	X	X	X				X	X
VIC PRIMARY CONTAINMENT SAMPLING & RECOMBINER LINES		X						X									X	X							
VIIA PCIG PROCESS LINES			X					X										X							
VIIB PCIG TIP PURGE SUPPLY		X						X										X							
VIIC PCIG TO ADS VALVES														X											
VIIIA DWCW/RECIRC CLG. WATER			X					X																	
VIIIB MISCELLANEOUS PROCESS		X						X																	
ECCS PROCESS LINES			X				X																		
BLOCK & VENTS		X						X									X	X							
INST GAS BLOCK & VENTS			X					X										X							
REFUELING FL. ISOLATION																	X		X						X
REACTOR ENCL. ISOLATION		X						X										X		X				X	

TABLE 2

ISOLATION SIGNALS

A	Reactor Level 3 - Low
B	Reactor Level 2 - Low, Low
C	Reactor Level 1 - Low, Low, Low
D	Main Steam Line - High Rad
E	Main Steam Line - High Flow
F	Turbine Enclosure - Main Steam Line Tunnel - High Temperature
G	Drywell High Pressure/Reactor Low Pressure
H	Drywell High Pressure
J	RWCU - dFlow, Area High Temp., Non-Regen. Htx. High Temp.
K	RCIC - Reactor Steam Line dPress., Exhaust Diaphragm High Pressure, High Temp.
KA	RCIC - Steam Supply Low Pressure
L	HPCI - Reactor Steam Line dPress., Turbine Exhaust Diaphragm High Pressure, High Temp.
LA	HPCI - Steam Supply Low Pressure
M	PCIG to Drywell - Low dPressure
P	Main Steam Line - Low Pressure
Q	Condenser Vacuum - Low
R	Refueling Area Ventilation Exhaust Duct - High Rad
S	Reactor Enclosure Ventilation Exhaust Duct - High Rad
T	Outside Atmosphere to Refueling Area - Low dPressure
U	Outside Atmosphere to Reactor Enclosure - Low dPressure
V	Reactor Pressure - High (RHR Valve Permissive)
W	North Stack Effluent - High Rad
Y	SLCS Initiation
Z1	Reactor Enclosure/SGTS Connecting Valves Failed Open (HV-76-196, 197)
Z3	Refuel Floor/SGTS Connecting Valves Failed Open (HV-76-019, 020)

presented to him, but draws incorrect conclusions from this information. For example, the cause of an automatic action could be misinterpreted or the appropriate corrective action in a situation improperly identified. These errors of analysis can be reduced by use of IRIS.

Plant procedures currently assist operators in diagnosing the cause of isolations. Procedures also exist which direct operators to reset or bypass isolations.(4) These procedures are useful and necessary. Replacing them with an expert system is not proposed or recommended. An expert system can, however, aid the operator in identifying the following:

* Procedures which must be used to bypass/reset the isolation.

* Reference Drawings which may be useful in determining the cause of the isolation.

* Specific plant instruments which may have caused the isolation.

* Specific valves which should be closed as a result of the isolation.

DEVELOPMENT TOOLS

The following section discusses the use of Personal Consultant Plus (PC PLUS) and dBASE III Plus as development tools for IRIS. Considerations in choosing these development tools are presented with an argument that compatibility with existing hardware and utilizing existing databases minimize development time and cost.

An Overview Of Personal Consultant Plus

Personal Consultant Plus (PC PLUS) is the software used to develop IRIS. PC PLUS is an expert system tool which has been used to build and deliver complex expert systems in many diverse areas.(5) It was written by Texas Instruments in the LISP dialect SCHEME. Users need not know LISP to use PC PLUS although SCHEME can be accessed through the LISP Edit facility. PC PLUS produces commercial expert systems primarily for a PC delivery platform but with the option of delivering knowledge bases to a mainframe environment.(6)

Knowledge Representation.

PC PLUS's knowledge representation consists of rules which can be written in LISP or PC PLUS's Abbreviated Rule Language (ARL). Rules (IF ... THEN ... statements) which are related to one another can be grouped into frames to allow economical access to knowledge without exhaustive searches of individual rules. Frames are a way of clustering rules based on their applicability at different stages of the problem solving process. The PC PLUS knowledge representation method also supports simpler rule-only systems that can be built by defining only one frame. Frames are related to one another through inheritance. Inheritance among frames increases programming efficiency because some relationships are implicit rather than

entered as rules. Meta-level knowledge improves the performance of the system by allowing the knowledge base to learn from experience which rules are most useful, causing the system to try the most useful ones first, and reordering the way rules are tried. Meta-rules can be written which allows system developers to fine-tune their inferencing strategies.

Chaining Strategies.

PC PLUS inference engine uses backward chaining as a primary problem-solving strategy. Backward chaining, goal driven inferencing, employes a depth-first search of rules to prove or disprove the current goal. Most frames contain a goal parameter whose value is sought through searching and testing rules which set the goal parameter. PC PLUS also supports forward chaining, which is data-driven. The inference engine moves from known facts forward to a goal which is not specified when the process starts. Forward chaining is specified for an entire frame or for individual rules by assigning the Antecedent property to the rules or frame. The combination of forward and backward chaining allows for an efficient design.

Interface To dBASE III Plus.

Utilities are available for interfacing external programs to PC PLUS. These include external DOS programs, Lotus 1-2-3, and of particular interest an interface to dBASE III Plus. This PC-based database management system (DBMS) by Ashton-Tate can be used to create databases whose data can then be manipulated by PC PLUS. PC PLUS can serve as either a back end system which uses information from the database, or as a front end system which selectively gathers information for the database. This access to a database is a powerful tool which can greatly extend the scope of an expert system.

Considerations In Choosing Tools

PC PLUS was chosen as the expert system tool for IRIS for the following reasons. Its primary development and delivery platform is a PC. This allowed development of the system using available hardware. PC PLUS can also be run on DEC VAX 11/780 machines, which were also available without equipment purchase. It is noteworthy that this type of portability preserves investment in development time if applications become too complex for a PC environment. A significant development cost savings is also realized by utilizing existing hardware.

The dBASE III Plus interface was utilized to minimize data entry while constructing the knowledge base. Much of the information needed for IRIS was already stored in database form. Loading this information into a dBASE III Plus database allowed much of the information needed by IRIS to be assembled into a knowledge base without manual data entry. This approach can significantly reduced the amount of time needed to develop a system, thereby reducing development cost.

DESIGN AND CONSTRUCTION OF IRIS

The following section discusses the general design strategy of IRIS and how the development tools were utilized in the design. The frame structure, inferencing strategy and dBASE III Plus interface are described. An example consultation is given. Details involved in conversion to an on-line system are mentioned. The benefits of building a system prototype and using a database interface are discussed. IRIS's current state of development is presented.

General Design Strategy

IRIS was designed to guide operators through the procedures for verifying, resetting, or bypassing isolations. It was also designed to aid in determining the cause of isolations. It accomplishes this by prompting the operator for **WHAT-HE-WANTS-TO-KNOW**, a defined PC Plus parameter which can be any parameter specified in Figure 2. Next, the system prompts the operator to find out **WHAT-HE-ALREADY-KNOWS.** Again, Figure 2 illustrates the possible parameter selections available to the operator. Now the system prompts the operator to find a value for the parameter **WHAT-HE-ALREADY-KNOWS.** From this information IRIS determines what the operator wants to know.

System Design

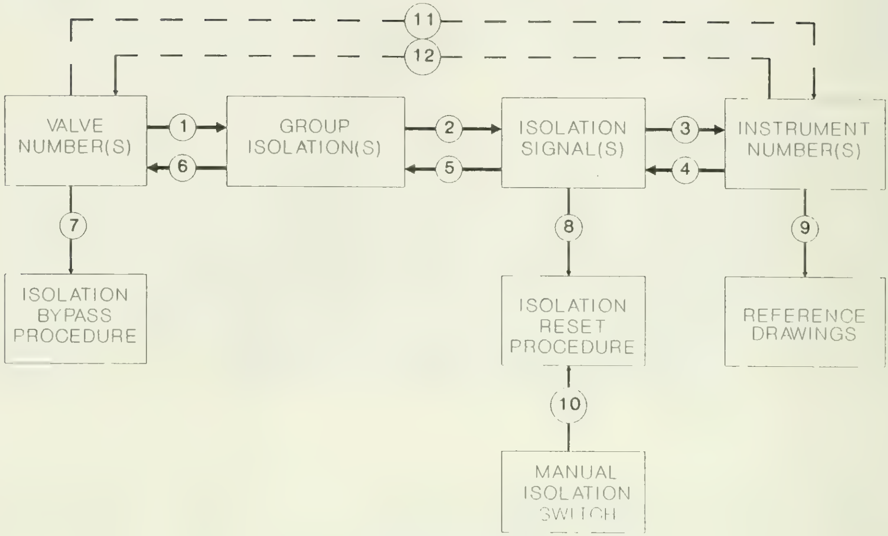
The root frame, or first frame which PC PLUS begins to instantiate during a consultation is appropriately named IRIS. This frame employs forward chaining to determine, through menu prompts, what type of parameter the operator wants to know and what type of parameter the operator already knows. Once the user has specified these two parameters, the rules in the root frame determine which subframe should be instantiated next.

The goal of the subframe which is instantiated next is exactly what the operator wants to know, that is, all rules in this subframe set **WHAT-HE-WANTS-TO-KNOW** to some value. Next, using backward chaining, a path is established from the current goal parameter, **WHAT-HE-WANTS-TO-KNOW**, to what is already known, **WHAT-HE-ALREADY-KNOWS.** During this process PC PLUS may invoke other subframes to determine needed parameters. These needed parameter are known as subgoals. After a rule is found which sets the current goal or subgoal, and the rule fires, a chain is established. This rule is able to fire because the information needed to set the rule was supplied by the operator. The following example will illustrate this process.

Assume a Group IIA isolation has occurred and the operator wants to know what instruments could have caused the isolation. Referring to Figure 2, **WHAT-HE-ALREADY-KNOWS** is that a Group Isolation has occurred and **WHAT-HE-WANTS-TO-KNOW** is what instruments could have caused the isolation. The system will prompt the user for this information and then will prompt to determine which Group Isolation has occurred. The frame which is instantiated next will have instruments as its goal and will contain rules which have the form:

```
IF ISOLATION-SIGNAL = A
THEN INSTRUMENTS = (LIST OF APPLICABLE INSTRUMENTS)
```

FIGURE 2



FRAME	WHAT-HE-ALREADY-KNOWS	WHAT-HE-WANTS-TO-KNOW
1	VALVE-NUMBER(S)	GROUP-ISOLATION(S)
2	GROUP-ISOLATION(S)	ISOLATION-SIGNAL(S)
3	ISOLATION-SIGNAL(S)	INSTRUMENT-NUMBER(S)
4	INSTRUMENT-NUMBER(S)	ISOLATION-SIGNAL(S)
5	ISOLATION-SIGNAL(S)	GROUP-ISOLATION(S)
6	GROUP-ISOLATION(S)	VALVE-NUMBER(S)
7	VALVE-NUMBER(S)	ISOLATION-BYPASS-PROCEDURE
8	ISOLATION-SIGNAL(S)	ISOLATION-RESET-PROCEDURE
9	INSTRUMENT-NUMBER(S)	REFERENCE-DRAWINGS
10	MANUAL-ISOLATION-SWITCH	ISOLATION-RESET-PROCEDURE
11	VALVE-NUMBER(S)	INSTRUMENT-NUMBER(S)
12	INSTRUMENT-NUMBER(S)	VALVE-NUMBER(S)

Since no Isolation Signal is known this will become a subgoal and the frame with Isolation Signal as its goal will be instantiated. This frame will contain rules of the form:

```
IF GROUP-ISOLATION = IIA
THEN ISOLATION-SIGNAL = (LIST OF APPLICABLE ISOLATION SIGNALS)
```

Referring to Tables 1 and 2, a Group IIA isolation can be caused by isolation signal A OR V (Reactor Level 3 - Low OR Reactor Pressure High). The system can now set **ISOLATION-SIGNAL** equal to A and V. Next, the instruments which generate Isolation Signals A and V will be determined. This information can now be displayed to the operator as the instruments which could have caused the isolation.

Conversion To An On-line System

Although IRIS has been designed as a conventional expert system, conversion to an on-line system could be accomplished using the Personal Consultant Online (PC ONLINE) software. Figure 3 illustrates a conventional expert system in which the operator notes the state of the process through observation and supplies the information to the expert system when prompted. The conclusions reached by the expert system are displayed to the operator who then evaluates the conclusions and takes appropriate control action. Figure 4 illustrates an on-line expert system in which the process is monitored by the expert system. Using PC ONLINE, IRIS could be converted into a system which would know when Isolation Signals had occurred and would automatically supply the operator with information pertaining to which instruments caused the isolation and what valves should be closed and other useful information. The benefits of this particular system having on-line capabilities would be most evident to the operator. With on-line operation prompting the operator for information would be minimized. The fact that PC PLUS supports on-line systems could be a significant advantage in many applications.

IRIS's Current State of Development

A system prototype or demonstration prototype was first developed which consisted of an expert system and database which could solve problems for the first three Group Isolations.(7) The database was constructed by manually entering data. Constructing a prototype allowed system structure to be developed before large number of rules made changing the structure difficult. The expert system structure was established and tested for the first three Groups and is now being completed to include all Groups. The database will be completed by downloading information from existing databases and assembling it into a dBASE III Plus database. This has been accomplished for much of the information by using the dBASE IMPORT command. This technique of using existing databases preserves work already invested in organizing information. Most utilities have vast quantities of information stored in databases which can be interfaced with expert systems by reconstructing the information in a database which can be accessed by the expert system. IRIS will reach the stage of research prototype once the database is fully assembled and additional rules are written to handle the remaining Groups.

FIGURE 3

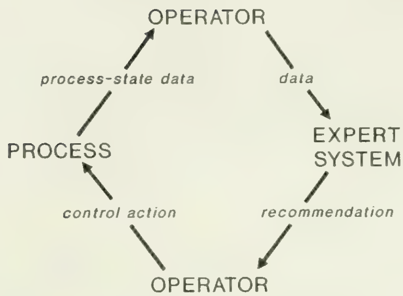


FIGURE 4

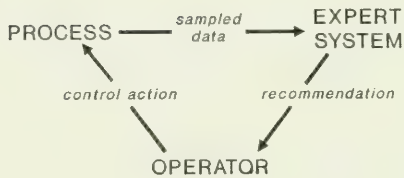


FIGURE 5

Benefits In Using A Database Interface

The speed of a database query is considerably faster than the inferential search of an expert system. Thus, performance of an expert system can be improved by providing a database interface in some applications. Another benefit is illustrated in Figure 5. Once a database has been constructed, any number of expert systems can access or update the information contained in the database. A new class of systems know as expert database systems (EDS) have been defined as systems for developing applications requiring knowledge-directed processing of shared information. A major advantage of this approach is the possibility of using existing databases to which expert systems can be connected as an application program.(8)

EXPERT SYSTEMS AND OPERATOR TRAINING

The following section discusses the use of expert systems as an aid in training nuclear plant operators. The ability of an expert system to explain how it reached a conclusion is discussed. It is argued that in some situations an expert system need not explain how conclusions were reached. In applications where advice is being given, the bases of the recommended action may be of more use to the user. The use of IRIS as a training aid is mentioned. The ability of IRIS to explain how it reached a particular conclusion is examined. Difficulties encountered in completing this aspect of the system are discussed. Use of expert systems in simulator training is also discussed.

Expert Systems As Training Aids

The use of expert systems as tutors or training aids is well documented.(9) As operators work with expert systems they are given extensive exposure to system facts, interrelations, and interactions. Exposure to this information is useful in reinforcing operator knowledge. If the system can also describe how it formulated a conclusion, further understanding can be gained. The ability of a system to explain its reasoning process will, in some cases, increase user acceptance of the system as well.(10)

In operator aid applications where the expert system is giving the operator advice on what actions should be taken, the ability of the system to state the basis behind the action is probably more useful to the operator than information pertaining to how the system reached a particular conclusion. Restated, the operator is probably more interested in why he should take an action than in how the system reached its conclusion. The programming involved in this type of system enhancement is simple and uncomplicated whereas the programming involved in developing a systems ability to describe the way it reached its conclusions is complicated and involved. The benefits gained by having a system explain how it reached its conclusions must be weighted against the increase in development time and cost.

Additional benefits are also realized by the developers of expert systems. The understanding gained in developing and encoding the knowledge base further enhances personnel training.(2)

IRIS As A Training Aid

IRIS's use as a training aid at this time is limited to increased operator exposure to the interrelationships between the Isolation Signals, Group isolations, valves which isolate and similar information. The systems ability to explain how it formulated a conclusion exists only in that it can identify which rules were fired during the consultation. This information is available through PC PLUS's option **HOW.** Although PC PLUS allows development of more advanced explanation facilities the programming involved is significant. IRIS's explanation facility will not be enhanced further unless it is necessary to gain user acceptance. As stated earlier, this ability is not always necessary and can be added at a later date if the original system is properly designed.

Simulator Training And Expert Systems

The use of expert systems in simulator training provides an opportunity for operators and management personnel to become familiar with expert system operation. The simulator environment is ideal because it presents the operator with actual plant transients and allows them to use expert systems while responding to transients. This not only benefits the operator who gains confidence in the expert system, it benefits the expert system developer who can assess how effective his system is in an operating environment. Management personnel can also be exposed to expert system in the simulator where their benefits can be easily observed. This is a critical step which must be taken before expert systems will be used in nuclear plant control rooms. Management must have the opportunity to see first hand the potential these systems have for improving operations. Expert system verification and validation can also be incorporated into simulator training. This will allow system bugs to be worked out before the system is put into service in the actual control room. Use of expert systems in the simulator provides an opportunity for introducing regulatory personnel to expert system applications. Demonstrating expert system operation to regulatory personnel must be done prior to deployment of any system giving advice to operators.

GAINING REGULATORY ACCEPTANCE

In this section, it is argued that the regulatory position on use of expert systems must be clear to utilities before most will pursue developing these systems. This issue is paramount in applying this technology to nuclear power applications. Other potential applications of expert systems to improve nuclear operations are mentioned.

Regulatory Position

Regulatory acceptance cannot precede regulatory personnel familiarity with expert systems. At the same time, many utilities remain hesitant to invest in expert systems without assurance that their use will be permitted by regulators. Without this assurance many expert system applications for the nuclear industry will not be used commercially but will remain as research prototypes. This situation must be addressed. A suggested solution is to first develop expert

system prototypes which do not require interface with plant equipment. This will provide regulatory personnel with an introduction to expert systems without raising the issues involved in deploying on-line systems. On-line systems which require plant modifications would probably be more difficult to implement due to regulatory concerns and lack of management support due to the aforementioned reasons. After successful deployment of an expert system which does not require interface to plant equipment then deployment of on-line systems can be seriously considered.

To summarize, the key to gaining regulatory acceptance is to expose regulators to expert systems and their benefits and then to request a formal position guideline pertaining to their use. Any such position statement would encourage utilities to develop expert systems which would enhance plant safety and reliability.

Other Potential Applications To Improve Nuclear Operations

Applications of expert system already being explored for use in nuclear operations include systems to track emergency operation procedures, alarm processors which would screen alarms and give immediate operator response and systems to analyze Tech Spec limiting conditions for operation. These systems have great potential for improving operations at all nuclear facilities.(2)

Other potential applications include expert systems which would analyze the effects of de-energizing electrical busses or equipment. Expert systems which aid in troubleshooting certain plant systems are also another potential application.(11) Equipment suppliers for the next generation of power plants should be encouraged to supply expert systems for troubleshooting their equipment. This will not only benefit the user who can expect to minimize equipment down time due to rapid problem identification, it will also benefit the equipment supplier by realizing a reduction in field service costs.

Much information used by operators and other plant workers is retrieved by looking through database printouts, complicated procedures, or lists of information. Much of this searching for information can be eliminated by storing the information in databases and providing an expert system interface to search the information. These user friendly database queries can save time and reduce duplication of work. The combination of an expert system and a database management system has innumerable applications in the area of power plant construction, operation and maintenance.

CONCLUSION

It is clear that expert systems offer the potential for improved nuclear operation. In particular, IRIS can improve plant safety by reducing operator error during analysis of automatic containment isolations. Additionally, personnel training benefits are realized through use and development of expert systems such as IRIS. While there are many application of expert system being developed for nuclear power few are being put into commercial use. Regulatory issues must be addressed before expert systems can be deployed for use in nuclear control rooms.

ACKNOWLEDGMENTS

The authors wish to acknowledge the contribution of David Kelsey for his work in compiling much of the information contained in IRIS. Thanks are also due to John Spencer, Edward Callan and Ronald DeGregorio for their assistance and support in this research project.

REFERENCES

1. Limerick Generating Station Final Safety Analysis Report.

2. R. E. Uhrig. "Nuclear utilities put artificial intelligence to work." POWER, June 1988, pp 35-38.

3. J. A. Shaw. "Operator Error: What Can We Learn?" ISA Computer Technology Collection, September 1986, pp 887-891.

4. Limerick Generating Station General Plant Procedures. " GP-8 Primary and Secondary Containment Isolation Verification and Reset".

5. S. J. Shepard. "Sophisticated Expert." PC TECH JOURNAL, July 1988, pp 106-119.

6. Personal Consultant Plus Reference Guide. Texas Instruments.

7. D. A. Waterman. A Guide to Expert Systems., Reading Massachusetts: Addison-Wesley, 1986.

8. J. M. Smith. "Expert Database Systems: A Database Perspective." Expert Database Systems: Proceedings from the First International Workshop, Menlo Park, California: The Benjamin Cummings Publishing Company, Inc., 1986, pp 3-15.

9. W. J. Clancey. Knowledge-Based Tutoring: The Guidon Program., Cambridge, Massachusetts: The MIT Press, 1987.

10. A. Basden. "On the Application of Expert Systems." Developments in Expert Systems., New York: Academic Press, Inc., 1984, pp 59-75.

11. K. R. Carr. "Expert systems applications for diagnosis and troubleshooting of computer systems and instrumentation." ISA Computer Systems Technology Collection, September 1986.

Thermal Performance Advisor Expert System Development

MICHAEL MCCLINTOCK
General Physics Corporation
6700 Alexander Bell Drive
Columbia, Maryland 21046, USA

RICHARD METZINGER
PSE&G
80 Park Plaza
Newark, New Jersey 07101, USA

NORRIS HIROTA
Electric Power Research Institute
3412 Hillview Avenue
Palo Alto, California 94304, USA

ABSTRACT

In recent years the electric utility industry has developed an increased interest in improving efficiency of nuclear power plants. EPRI has embarked upon a research project RP2407, "Nuclear Plant Performance Improvements" which is designed to address needs in this area. One product of this project has been the "Thermal Performance Diagnostic Manual for Nuclear Power Plants" (NP-4990P). The purpose of this manual is to provide engineering personnel at nuclear power plants with a consistent way in which to identify thermal performance problems.

The next phase of that project, which is being developed by General Physics Corporation under the joint sponsorship of EPRI and Public Service Electric & Gas Company, is the development of the Nuclear Thermal Performance Advisor (NTPA), a computer system designed to make the kinds of information contained in the Thermal Performance Diagnostic Manual (TPDM) available to the performance engineer "on-line." The NTPA will considerably extend the TPDM in that it has an expert system component that will aid the engineer in making a detailed analysis of the plant and any sources of thermal inefficiency.

General Physics is also involved in the development of another computer system called Fossil Thermal Performance Advisor (FTPA) which helps operators improve performance for fossil power plants. FTPA is a joint venture between General Physics and New York State Electric & Gas Company. This paper describes both of these computer systems and uses the FTPA as an interesting comparison that illustrates the considerations required for the development of a computer system that effectively addresses the needs of the users.

NTPA HARDWARE and SOFTWARE TOOLS

The initial plan was that the NTPA would be built using tools that were available at the time the project was initiated. These were the IBM AT running under DOS and using software tools that were available in early 1987. It was recognized, however, that in the area of computer hardware and software introduction of improved tools progresses at such a rate that one must be careful not to be so conservative in the selection of tools for a long term project that they are obsolete by the time the system is completed.

Accordingly, when work on the project began in late 1987, a review of the available tools was conducted as the first task in the project. A careful investigation of both software and hardware was conducted. Our objective was to use tools that are powerful enough to provide both for the job at hand and future expansion on one hand, while being proven, readily available, and as inexpensive as possible on the other hand.

The result of that review was the following:

- The computer used is a Compaq 386/20 with an 80387 math coprocessor, 16MB random access memory, a 60 MB hard disk, a 5.25 inch 1.2 MB floppy disk drive and a 60 MB tape drive. This choice was consistent with the desire to build the system using a "PC type" computer which is familiar to the industry and yet powerful enough to provide for the needs of this project and expected later expansion.

- The Unix System V operating system was chosen to avoid three major limitations of DOS. Most importantly is provides for virtual memory, freeing us from the 640K limitations of DOS. Second it provides for multitasking which provides the capability for a real time system and greater flexibility in design of the software architecture. Third, Unix provides multiuser capability, thus the system may be made available to a number of users at different locations around the plant, and even remotely.

- Nexpert Object is the expert system tool that is being used for the expert system component. Nexpert Object is a hybrid rules and objects-based expert system development environment which run on UNIX™ computers, DEC VAX workstations and many other computers. NEXPERT is written fully in the "C" programming language, insuring a high level of portability, integration, and performance. It includes capabilities for integrated forward and backward chaining, pattern-matching, and all other capabilities required to develop the knowledgebase.

  NEXPERT's comprehensive graphic development interface allows the development team to edit rules and objects as well as build control structures. The open, event-driven architecture permits the development of real-time, on-line applications and full communication and interaction with the task environment. These capabilities allow for the future expansion of the system. NEXPERT rules and objects can directly communicate with the database module, retrieving data at any time during program execution. The runtime system allows the delivery of versions of the system on any computer which runs UNIX™.

- DVTools by VI Corp. and X-Windows are used for graphics in NTPA. The monitor for the system is a 19 inch, high resolution, color-graphic monitor driven by an X-Windows compatible graphics board. Like Nexpert, DV Tools is written in C and is portable across a wide range of hardware platforms. This, together with the X-Windows standard for graphics under Unix, assures that the NTPA system will have the greatest possible flexibility and portability.

- At the time this project was begun not all of the software tools that were chosen were available for the Compaq/Unix platform. It was determined therefore that development of the system would be started using an Apollo Domain Series 3000 engineering workstation. A prototype of the system was developed on that platform. The system is now being ported to the Compaq/Unix platform.

NTPA Design Description

NTPA is being developed as a aid for the performance engineering personnel at PSE&Gs Salem Nuclear Plant. It is designed to be an off-line system that will reside in the engineer's offices and be available at any time to analyze the state of the plant.

The Salem Nuclear Plant has two nearly identical units. They are Westinghouse PWRs rated at 1162 MWe. Unit 1 was started up in 1977 and unit 2 in 1981.

Plant performance is monitored on a daily basis by the performance engineer. He does his analysis based on a review of about 200 analog points from the plant computer and a "walk around" of the plant. The plant computer has no facilities for remote, automated access and all data is manually downloaded.

In the design of the NTPA computer system it was necessary to keep in mind the environment in the plant and the available resources. A system that was built on the presumption of availability of nonexistent instrumentation would be of little value.

In broad terms, the NTPA system is an interactive heat rate diagnostic expert system that provides nuclear power plant thermal performance engineers with the capability to identify and correct causes of lost power generation. In general engineers can:

- Accumulate performance data for subsequent retrieval and analysis.

- Obtain expert advice which identifies probable causes of lost power generation.

- Print a summary report that details performance calculations and the state of the plant for any stored data set.

These functions are achieved using modular software components, shown in Figure 1, which are organized as follows:

<u>NTPA Database Component</u>. The database component provides the "link" between the NTPA and plant data. The NTPA reads data from DOS diskettes which contain data downloaded from the plant computer. This data is stored in the NTPA's own internal database. Once the data is stored in the NTPA it can be retrieved and analyzed at any time, although it is expected that the analysis of the most recent data set will be most often done. The database programs also perform necessary data management functions, such as archiving data to tape for long term storage and maintaining various constants required for performance calculations.

<u>NTPA Parameter Calculation Component</u>. This component performs the engineering calculations necessary for evaluating plant performance. In general three types of calculations are performed. <u>Composed</u> calculations compute values for parameters which are based on actual plant data, for example an average of several thermocouple readings or a heat transfer coefficient. <u>Target</u> calculations determine the best achievable value for either an actual or composed variable. <u>Evaluation</u> calculations convert quantitative data to the qualitative data required in the expert system component. The result of these calculations are called EVALs.

<u>NTPA Expert System Component</u>. This component includes the Nexpert^{tm} kernel and plant specific knowledge bases. The Nexpert^{tm} kernel draws conclusions about plant problems using data from the NTPA database component and calculated parameters, and the knowledge bases. This component also contains the programs necessary for linking the NTPA to the Nexpert^{tm} kernel and interpreting the results of its diagnoses. NTPA has a control knowledgebase that identifies the most likely cause of lost generation. When the control knowledgebase identifies a likely area for further investigation, it calls up the appropriate diagnostic knowledgebases. The areas initially covered by plant specific diagnostic knowledgebases include the following:

- Moisture Separator Reheaters
- Condenser Backpressure
- High Pressure Feedwater Heaters

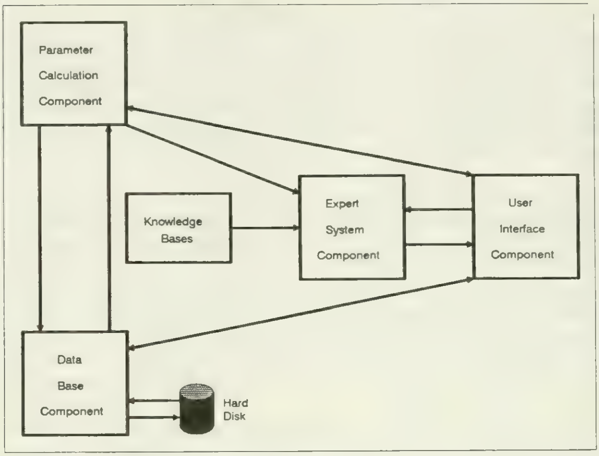


Figure 1. NTPA Software Architecture

There are a number of other diagnostic areas covered by the TPDM for which plant specific knowledgebases have not yet been developed. These areas have less complete generic diagnostic knowledgebases which are derived directly from the TPDM.

NTPA User Interface Component. The user interface programs provide the means for the user to interact with the NTPA. These programs provide the graphics and text required by the expert system component to explain its diagnoses. Further, these programs allow the user to execute various commands or enter input into the system. The system uses a mouse (a pointing device that enables the user to make selections from the screen by "clicking on" various icons) for control of the screens and quick access to the system.

NTPA Functional Description

The NTPA system was designed to suit the requirements of the Salem plant. It is important in the design of any computer system to assure that a thorough understanding of the plant and its environment, both organizationally and physically, be gained first. The design must then reflect the plant and its needs. The following describes the typical use of the NTPA system.

A technician will manually download data from the plant computer using an interface which is equipped with a modem. The data will then be downloaded from the interface to a DOS computer via modem. The data files will then be copied from the DOS computer to a floppy disk which will then be loaded into the NTPA computer.

The NTPA will employ an easy to use interface to read the floppy disk containing the plant data into its data base. There is data required for analysis of the plant which is not available from the data provided from the plant computer on the floppy disk. The NTPA system will provide for easy entry of the required data which will include such information as identification of which circulating water pumps are running. In general, where manual entry of data is required, the plant engineer will be prompted to edit a default value. Thus, the system will be able to perform an analysis of data using default values if the plant engineer elects not to enter actual data.

Following the entry of the data, the plant performance engineer can initiate an analysis of the data. The plant parameter calculation component will calculate parameters necessary to produce a summary report of the unit status showing the target load as compared to the actual load and any known causes of lost load, such as circulating water pumps not in service.

If the plant is in normal operation with all components, such as circulating water pumps, in service and there are no problems, the actual load and the target load will be the same within a normal tolerance. In the event that there are some components, such as circulating water pumps out of service, there will be an identified loss of load. If there are no sources of lost load other than those identified from the plant data, the sum of the known loss(es) and the actual plant load will equal the target plant load within a normal tolerance.

In the event that the actual load (or actual load plus known losses) are less than the target load by more than the normal tolerance, the expert system component will automatically prompt the performance engineer to initiate a diagnosis.

The expert system uses modular knowledgebases. One of the modular knowledgebases is the control knowledgebase will be called up and run every time a diagnosis is initiated. The control knowledgebase will be used to identify the diagnostic area that is the most likely cause of lost load for investigation in detail. This approach to problem solving is similar to the manner in which a human expert might approach problem solving. Once the control knowledgebase has identified the diagnostic area for investigation, it will load the modular knowledgebase for that diagnostic area.

When a modular diagnostic knowledgebase is called up, the evaluation/diagnosis will generally be done in two stages. First a diagnosis using only the data already entered into the database will be performed. In most cases, that diagnosis will not be complete and thus unsatisfactory because additional data which must be gathered manually will be required. Typically the type of information that will be required is readings from local pressure gauges and the positions of manual valves. In this first stage then, a preliminary diagnosis will be made, and a listing of additional data required for a more complete diagnosis will be printed out for the performance engineer.

The performance engineer normally makes a tour of the plant once each day to review plant status. The printed report from the stage 1 diagnosis will provide him with a list of data required that he can gather in the course of that tour. When he returns from the tour, the additional data can be entered and the second stage diagnosis performed. In general, it is expected that additional data will provide for a better diagnosis (more specific and having a higher degree of confidence) than the first stage diagnosis.

It is expected that even with additional data that is available from the daily plant tour, the system may not be able to arrive at a conclusive diagnosis. In such cases, the system may recommend specific tests to further identify the source of the problem. As an example, a simple test to determine feedwater pressure drop through a feedwater heater may be recommended to confirm or disprove the possibility of a leak in or bypass around a feedwater heater waterbox partition plate.

When the system completes a diagnosis, the screen will automatically switch from the default screen to one that shows a subdiagram of the plant appropriate to the diagnosis (if high condenser pressure is identified as the cause of lost load, for example, the condenser/circ water system subdiagram will appear), the most likely diagnosis, the rule for the most likely diagnosis, and a summary of all possible diagnoses. The user will have the option of displaying a "logic tree" in place of the plant subdiagram. This logic tree will illustrate the "reasoning" of the system and

provide the performance engineer with the ability to query the system. A hard copy summary report of the diagnosis will be available on demand.

The performance engineer will be able to "browse" the system using a mouse to select first a PLANT VIEW screen, and then lower level screens arranged hierarchically and selectable by selecting icons. He will be able to display parameter vs time plots (for the last ten data samples) of parameters represented by instrument icons on the graphic screens. Note that since the data is inputted manually at intervals that are not uniform, these plots may have gaps.

FOSSIL THERMAL PERFORMANCE ADVISOR

At about the same time that work began on the NTPA, General Physics and NYSEG began development of the Fossil Thermal Performance Advisor (FTPA). Considerable work in the area of computer performance monitors had been done by General Physics and NYSEG before undertaking this project. A controllable parameters monitor called Thermal Information Program (TIP) was developed by GP and then enhanced under the sponsorship of NYSEG. This system lent itself well to further enhancement by addition of an expert system component. A prototype system called X-TIP (for eXpert TIP) was developed both to determine the feasibility of the system and to learn what should be considered in the building of a full scale system. The prototype was successful and as a result, it was decided that we would proceed with the development of the FTPA system.

FTPA employs substantially the same hardware and software tools as those used in NTPA. There are significant differences in the two systems, however. Where the principal user of the NTPA system is the thermal performance engineer in a off-line situation, FTPA is designed principally to address the needs of the fossil plant control room operator for an on-line monitoring system with an expert system component to assist him in identifying operator controllable losses and how they should be corrected. It was just as important to learn the Somerset plant and its needs before designing the FTPA system as it was to learn about the Salem plant before the NTPA was designed.

FTPA Design Description

The system is interfaced with the Leeds & Northrop plant computer system through a universal data buffer which "appears" to the plant computer to be a printer, and to the FTPA system as a terminal. Currently about 750 data points are "dumped" every 5 minutes. This approach to interface with existing plant computer systems means that any plant computer that can be made to produce a printed report an a regular basis can be easily and inexpensively interfaced with FTPA. Figure 2 shows the physical arrangement of the FTPA system expected by the end of development.

The software architecture of FTPA is shown in Figure 3. FTPA has several inter-dependent software components which are run as separate tasks under Unix. These components perform the following functions.

- Component 100 - Expert System Component - This component consists of the expert system shell, the knowledgebase, and the interface which is C code which conveys data into the knowledgebase and conclusions from the knowledgebase.

- Component 200 - Plant Parameter Calculation Component - This component performs calculations both for the monitoring portion of the system and the expert system component. In general there are three types of parameters determined, targets for various performance related parameters, EVALS (numerical data which has been converted to the symbolic form for use by the expert system), and costs. This component is also where the plant model resides.

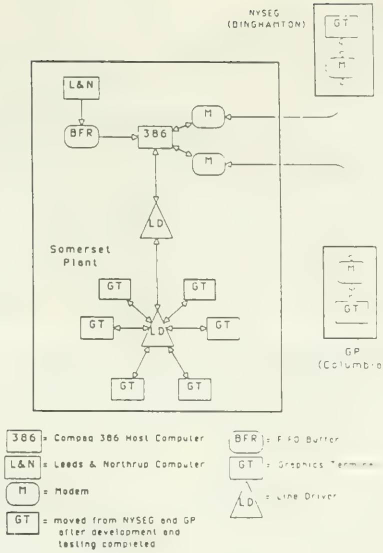


Figure 2. STPA System Hardware

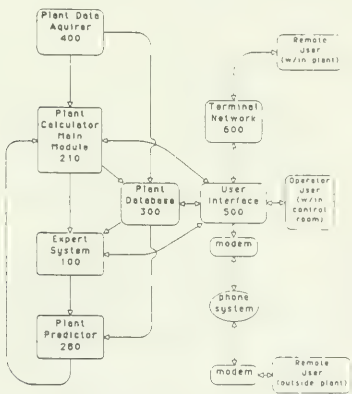


Figure 3. STPA System Design

- <u>Component 300 - Plant Database</u> - This component formats and stores the measured and calculated values of the variables monitored and produced in FTPA.

- <u>Component 400 Plant Data Acquirer</u> - This component interfaces the L&N plant computer with the FTPA computer.

- <u>Component 500 User Interface</u> - This component provides the graphic, keyboard and "mouse" interface that enables the operators to work with FTPA.

- <u>Component 600 Terminal Networker</u> - This component provides for communications between the control room computer and the remote terminals.

FTPA Functional Description

The FTPA provides a controllable parameters monitoring screen that provides the control room operator with target values for about 12 operator controllable parameters and the cost for off-target operation for each of these parameters. In addition to the controllable parameters screen, it provides a number of graphics screens that are simplified schematics of the plant and its various systems which represent plant components and their instrumentation as icons.

On each screen the instrumentation that is important for performance monitoring is represented by an icon. The user can display a strip chart emulation for any of these parameters in one of the three "windows" at the top of the screen by selecting the icon (clicking on the icon) with the mouse. The last 10 samples of data, both actual and (where appropriate) target, are displayed for each parameter.

The graphics screens are arranged hierarchically. That is, for example, the user may select a view of the combustion air and flue gas system by using the mouse to "click on" any of the components of that system shown on the plant overview graphic screen. He might then choose to see a detailed schematic of a particular air heater by clicking on the air heater in that new schematic.

The expert system component evaluates the data in each sample and makes a determination as to whether or not a diagnosis is required for a controllable parameter on the basis of cost. More than one diagnosis, up to three, may be made for a given data sample. The operator is alerted to the diagnosis by an annunciation which gives a brief description of the results of the diagnosis. The operator may simply acknowledge the diagnosis, or he may ask for more information.

If the operator asks for more information on a diagnosis, the screen is changed to show a simplified schematic of the area of the plant appropriate to the diagnosis, a more complete statement of the diagnosis, and any other possible causes for the problem diagnosed. The operator can select a "logic tree" which graphically illustrates the expert system's reasoning for the problem. Explanations associated with each of the rules in the logic tree are available to the operator so that if he is not satisfied with the diagnosis provided by the system he can easily investigate it further himself.

Note that while considerable information is available to the operator, nothing is "forced" on him. The system continually "looks for" opportunities to improve unit performance and provides an annunciation to alert the operator when one is found. The operator can then use the available information at his discretion.

Another feature which we plan to build into FTPA is a plant model. Most controllable parameters monitors operate under the assumption that a given mode of operation of the unit based on rules of thumb is the most efficient. FTPA would use the plant model to evaluate the current mode of operation to determine whether or not there is a more efficient mode available resulting from tradeoffs in losses, availability of equipment or deterioration of components. One example of this might be an evaluation of the trade-off between increased combustion air for the boiler to reduce unburned fuel loss and maintain steam temperatures at low loads (improves efficiency) and increased stack losses and auxiliary power consumption (hurts efficiency).

FTPA is a multiuser system. Remote terminals can be located in engineering offices in the plant or even corporate offices many miles away via modem connection. These remote terminals can be used to see the same information that the operator has available to him. Recently only text, rather than colorgraphic remote terminals were available in the market. It is anticipated, however, that new developments in graphics software standards and hardware will make it possible within the current year to obtain remote colorgraphic terminals to display exactly the same screen as that in

the control room. These remote terminals will also provide engineering and management personnel with the ability to do analysis of past data, including "what if" scenarios in which the engineer can "ask" the system what would happen if certain key variables were to be changed.

FTPA is being developed and implemented in NYSEG's Somerset plant (a 625 MW coal fired unit) in three phases. The first phase, which has been in operation in the plant since July, 1988, is an on-line controllable losses monitoring system. The second phase, which incorporates the expert system diagnostics, was installed in March, 1989. The third phase, incorporating multiuser capabilities, statistical functions, and perhaps the plant model, will be installed in early 1990.

Comparison of NTPA and FTPA

Both NTPA and FTPA are computer systems designed to aid power plant personnel in monitoring and improving plant heat rate. As such there are many similarities. There are, however, many differences as well. A review of some of the differences provides insight as to the importance of considering the user and his needs in the design of the system.

- On-line, real time vs off-line - Perhaps the greatest single difference between the two systems is in this area. In general the demands of a real time system are much greater than for an off-line system. Multitasking software architecture is an absolute requirement in the real time system to permit the user to interact with the system without interrupting its monitoring and data collection function. It also requires that the system functions operate fast enough to be completed in one time cycle. Finally, a real time system that provides for user interaction must generally provide for some functions to operate synchronously (keep pace with the plant computer which is periodically sending data for analysis) while others operate asynchronously because they are being "driven" by user interaction at the pace of the user.

- Availability of data Both the NTPA and FTPA systems are designed to function with incomplete data. The result of incomplete data is generally a decline in the quality in the diagnosis, in terms of the specificity and degree of confidence in the diagnoses. Clearly the more and better the information available the better the diagnosis can be. We discovered that the Salem plant has much less information available from the plant computer than the Somerset plant. This is principally because the Somerset plant is relatively new as compared to Salem and thus reflects the state of the art in place about 15 years later than that at Salem. Thus at Somerset almost all of the data required for diagnosis is available automatically from the plant computer. Some data must be input manually, however it is generally rather few pieces of data.

At Salem on the other hand, much of the data required for analysis is only available by gathering data manually. Thus in the Salem system we are forced to build provisions for easy entry of large amounts of data. These provisions include the generation of lists of required data and a spreadsheet style data entry in which the user has the option of accepting default values or entering actual values.

- User Interface - There are many similarities in the user interfaces of NTPA and FTPA. There are however, many differences as well. These differences are dictated by two factors, the intended user and whether or not the system is real time. In the FTPA system, one of the principal functions of the system is monitoring with no action from the

expert system component. The operators rely heavily on the monitoring functions. We designed the user interface so that when the expert system is invoked it alerts the user, but does not obscure the monitoring function (by "taking over" the screen) unless the operator chooses to do so.

In contrast, it is expected that the performance engineer will want to use the expert system component every time that he uses the system. He has no real time monitoring function to be concerned with. In NTPA, then, the expert system is allowed the use of the entire screen.

Customizing NTPA - Technology Transfer

The goal of the NTPA project is to develop a software kit that will facilitate the customization of NTPA for other EPRI members' plants. It should be understood that while considerable effort has been devoted to developing a system that can be customized with the least possible effort, the state of the art is such that even the least possible effort is non-trivial. NTPA is not a piece of generic, "shrink wrap" software. It is a complex computer system. In order to be useful in a given plant, it must be customized to the specifics of that plant. The areas of customization are as follows:

- Database - An expert system needs data in order to function. Most plants have many data points (from 200 - 1000) available through a plant computer or data logging system. The exact nature and naming of those points is different for each plant. It is necessary to "map" the plant data into the NTPA database.

- Engineering calculations - The engineering calculations required for most plants will not differ much in principle. There will be differences in detail dictated by the configuration of the plant however. For instance, the same basic calculations for target condenser pressure are applicable to most plants, however some plants have two zone condensers which operate at different pressures. In such a plant it will be necessary to calculate two different target pressures rather than only one.

- Expert system - Most of the types of problems that occur in power plants are similar. This implies that the general structure of the knowledgebases will be similar. The details of the knowledge bases will depend on the plant specifics. These specifics include the same sorts of differences in plant configuration that affect the engineering calculations. These specifics also include differences in plant operating environment (consider two otherwise identical plants, one of which is located in Maine while the other is located in Florida).

- User Interface/graphics - Both NTPA and FTPA make heavy use of high quality graphics. These graphics are necessarily plant specific. In order to assure enthusiastic acceptance by the intended users, the user interface should be built in an interactive fashion in which involves the users. It might be expected that a generic interface would generally not be well accepted by the intended users.

Future Plans for NTPA

In the course of development of NTPA to date, we have identified a number of areas for further work. These include the following:

- Develop and present a training program for EPRI members to enable them to develop a plant specific NTPA. This program will be designed to supplement a "software kit" consisting of a detailed user's manual, programmer's manual, and "example" software developed at Salem under the current workscope for NTPA.

- Develop six additional modular knowledgebases to cover those diagnostic areas identified in the TPDM and in other work, but not currently covered by plant specific knowledgebases in NTPA.

- Incorporate a plant model to provide the performance engineer with capability to do more detailed quantitative analysis of the plant, including quantifying the effects of known performance problems and optimization of the plant in off-normal operation.

- Currently NTPA is designed to function at or near full load only. Improved functionality would result from extension of the plant calculations and knowledgebases to provide for operation at low loads.

- Customize and install NTPA at beta test sites, including at least one BWR plant.

Conclusion

An expert system is not an end unto itself. The true usefulness of an expert system only becomes apparent when it is part of an integrated computer system. A successful system, that is a system that is useful for and used by the intended users, requires careful consideration of and design to accommodate the needs of the user and the environment in which the system will function. A corollary to this statement is that for a system like NTPA to be useful, customization to the plant in which it will be used is a necessity. Our experience to date demonstrates that through careful system design it is possible to build a system that can customized to the needs of specific plants with a reasonable effort. Our long term efforts will be directed towards making this technology available to as many EPRI members as have an interest in taking advantage of it.

Feedwater Heater Life Cycle Advisor: An Expert System Application for Nuclear Power Plants

STANLEY H. LEVINSON
The Babcock & Wilcox Company
3315 Old Forest Road
P.O. Box 10935
Lynchburg, Virginia 24506-0935, USA

ABSTRACT

This paper describes a methodology that could be used to develop an expert system life cycle advisor for feedwater heaters in nuclear power plants. Specifically, the objective is to develop a personal computer (PC)-based tool to assist utility personnel in using the guidance, methods, and good practices that have been previously developed and used for the operation and performance of feedwater heaters.

Seven steps in the methodology have been identified:

- Develop a list of direct and indirect activities associated with feedwater heaters
- Collect data related to feedwater heaters
- Select hardware/software
- Define knowledge/database structure requirements
- Define analytical/processing requirements
- Define interactive user interface requirements
- Develop the Feedwater Heater Life Cycle Advisor (FHLCA).

The final package will assemble the knowledge of experts, as well as information contained in various reports, into a software tool that can easily be used by a variety of utility personnel to assist them in decisions concerning feedwater heaters.

INTRODUCTION

Nuclear power plants represent complex interrelationships between many systems and components. As a result, plant personnel are unable to read and absorb the large amounts of technical information and data regarding the operation and performance of key components.

With the increasing capabilities in personal computer (PC) hardware (e.g., more disk space and memory, increased speed and throughput, etc.) coupled with the advances in available software (e.g., graphics, database management system, expert systems and "shells"), a computerized system can be developed that permits utility personnel to benefit from the vast amount of information that has been collected, but, not integrated.

OBJECTIVES

This paper describes a methodology for developing a PC-based tool to assist various utility personnel (operations, maintenance, engineering, etc.) in using the guidance, methods, and good practices previously developed for the operation and performance of feedwater heaters. A Life Cycle Advisor (LCA) addresses all phases of a feedwater heater's life -- from preventive maintenance to unscheduled repairs, from increasing optimum performance to making economic decisions, such as repair-or-replace. This decision and information tool should assist the plant personnel in minimizing costs during the feedwater heater's life, and, additionally, serve as a training tool for all elements of the life cycle (i.e., procurement, operation, maintenance/repair, diagnostics/prognostics, and failure analysis).

Since an LCA must be useful to the prospective user, it should be developed within the framework of those items most important to utilities: operations (availability) and performance (efficiency). While considering items important to feedwater heater's life (e.g., when to replace a tube bundle), a Feedwater Heater Life Cycle Advisor (FHLCA) should also address performance (e.g., what is the optimum level) and operation (have the level alarms been reset to correspond with the new level) considerations.

The basic approach is to determine potential users and their needs. Once these are defined, inputs, outputs, and analytical procedures can be specified. The nature of the inputs and outputs will determine the user interface and knowledge/database requirements.

This paper discusses the basic steps involved in developing of an expert system life cycle advisor. These tasks are briefly listed below:

1. Compile of all the direct and indirect activities associated with a feedwater heater. The appropriate utility users for these activities can be identified so that the necessary input and output for each activity can be determined.

2. Collect data related to feedwater heaters. This step involves identifying and sorting data. Information and knowledge will come from reports and individuals (i.e., experts).

3. Determine the software/hardware platform. It is anticipated that an FHLCA will be able to run on an IBM/AT (or equivalent) with 640K of memory.

4. Determine the structure for a conventional (traditional) database, as well as specifications for an AI-based knowledge base.

5. Determine algorithms that can be encoded in a high level language (e.g., C). In addition, specify the necessary AI-paradigms, where heuristic and/or non-deterministic decision making is necessary. The AI-paradigms apply to both the feedwater heater decisions and the menu interface with the user.

6. Define requirements for the user interface; this interface comprises of a "standard" and "smart" interface. The "standard" interface will include a hierarchial menu structure with various input methods. The "smart" interface will guide the user through an FHLCA, requesting more information where incomplete or inconsistent, determining which analytical methods to use, etc.

7. Enter data into the knowledge/ database(s), write the software (both traditional and AI-based), and test the package. A rapid prototyping strategy is recommended during development.

FUNCTIONAL DESIGN

The functional design for an FHLCA is presented in Figure 1. The figure shows a group of utility users, including Engineering, Maintenance, Operation, and Management; a fifth user function, Training, indicates that an LCA can be used as a training tool, as well as in aiding the decision making process.

Actually, the presented functional design could represent an LCA of any component. The basis for this structure is to satisfy the user's needs, that is, identify problems, determine input requirements, solve problems, and determine output format. These user actions can be accomplished via the "standard" and "smart" interactive user interfaces. These interfaces can consist of menus, windows, queries, graphics, and the like, and will permit access to analytical models and data.

The analysis tools are shown in two separate subsets. The first, the analysis base, represents the solutions that can be obtained using conventional computer software technology, i.e., writing a program in C. These techniques are typically used for problems that have algorithmic solutions. The second box represents the use of advanced programming techniques (forward-chaining, backward-chaining, heuristic searching), i.e., use of "expert systems." These techniques are generally used with problems that have no algorithmic solution, but rely on "rule-of-thumb" or heuristic types of solutions. Expert system techniques can also be useful in tracing procedures.

The last box on Figure 1 shows the Knowledge Base (KB). The KB consists of all the data and information that are required to perform the conventional and advanced analyses. The KB is broken into two parts: conventional database and information stored in a form compatible for various Artificial Intelligence (AI) paradigms. These AI techniques include object-oriented programming (requiring object definitions, attributes assigned, inheritance, etc.) and the use of production rules. The method by which information is placed into the Knowledge Base depends on how it is to be used by the analysis routines.

OPERATIONAL DESIGN

The operational approach of an FHLCA is presented in Table 1. The first column indicates general steps, which could apply to an LCA for any plant component. The second column, where appropriate, gives specific examples for feedwater heaters.

LIFE CYCLE ADVISOR DEVELOPMENT DESCRIPTION

The steps are presented in a sequential fashion, though in actual performance, there is a great amount of parallelism between tasks. A technique known as "rapid prototyping" can be used to develop the software for an FHLCA. Using this method, a small cross-section of the problem is focused on, and completed from beginning to end. The data necessary for one activity can be determined and placed in the database. The analysis requirements for that activity can be determined and coded. The necessary menus and user interfaces for that activity can be implemented.

The advantage of this methodology is that after a short period of time, the program can be functional and tested by a target user. This allows the developer to

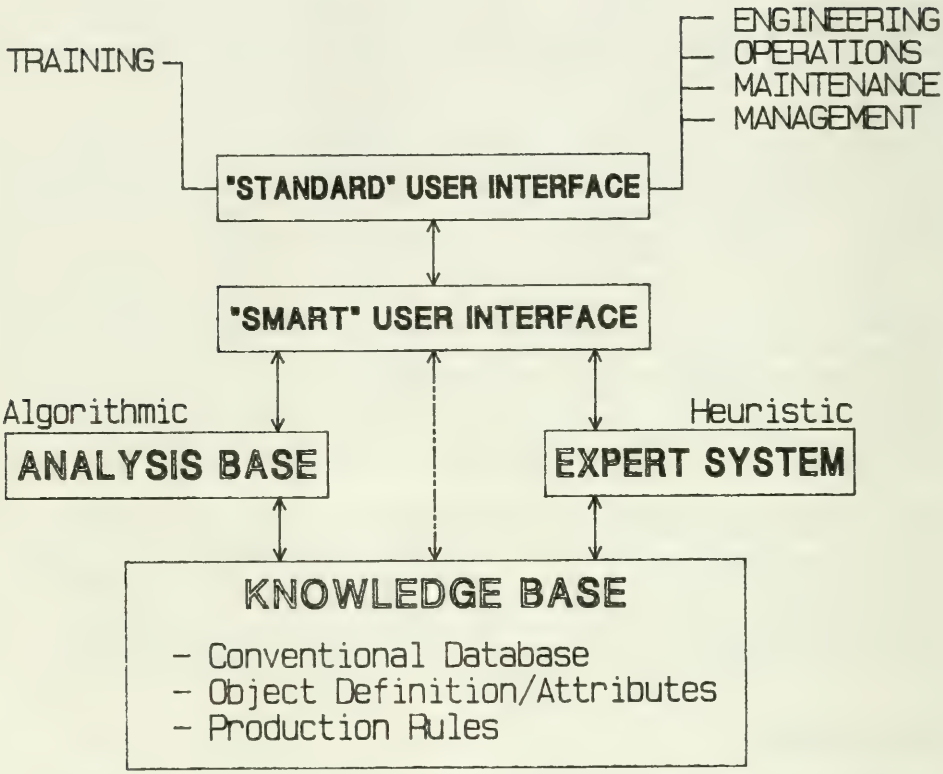


Figure 1 FUNCTIONAL DESIGN FOR FEEDWATER HEATER LIFE CYCLE ADVISOR

Table 1

GENERIC AND FEEDWATER HEATER CONSIDERATIONS OF OPERATION

GENERIC	SPECIFIC
1. The user specifies the problem to be solved via the menu structure available in the user interface.	1. When should feedwater heater tube bundles be replaced?
2. The Analysis Base and/or Expert System program determines what data (knowledge), i.e., inputs, are required to solve the requested problem.	2. How many tubes are plugged in existing unit? What has the plugging trend been with this unit. What is the required downtime to replace the tube bundle?
3. The need information is extracted from the Knowledge Base.	3. Not applicable.
4. The Analysis Base and/or Expert queries the user for any missing data or information. If appropriate and/or requested, the new information is stored in the Knowledge Base.	4. When are the upcoming planned outage? What is the lead time for a replacement tube bundle.
5. The Analysis Base and/or Expert System uses the gathered information to solve the problem.	5. Not applicable.
6. The solution is made available to the user via the user interface. The user may have a choice in the way the answer is presented (i.e. tabular or graphical).	6. Replace the tube bundle at the next scheduled refueling outage. User could try different scenarios (i.e. use different input data) to examine the sensitivity of the solution. The user, not the LCA, must make the final decision.

"rapidly prototype" a feasibility demonstration that will be representative of the entire program. During this process, lessons will be learned (through feedback with potential users) that will permit the development team to refine and extend the existing software, and use that knowledge during the rapid prototyping of the next "slice."

Determine User's Needs

Prior to the data collection process, specific users profiles (i.e. likely users of an FHLCA) must be identified. In a parallel effort, any activities (analyses, decisions) related to feedwater heaters that the utility performs can be determined. With this information, the desired outputs, required inputs, and analytical procedures can be defined.

The generic life cycle of a hardware component can be divided into procurement, design, installation, operation, maintenance, replacement, and training. A compilation of all the direct and indirect activities associated with a feedwater heater during its life cycle can be developed. Guidance for determining these activities will come from discussions with utility representatives, as well as previously published reports.

This compilation will become the impetus for organizing the data and analysis processes, determining additional data required, and screening and development of the topic/data information base of the hardware component. As the data search and screening are performed, the need for additional activity topics or further subdivisions will become apparent.

The required output for each potential user (as appropriate) for each identified activity can be determined. With the output defined, the necessary inputs can be determined. The inputs represent the information that can be stored in the Knowledge Base. Since the nature of the inputs may be vague, this step can be performed in conjunction with collecting data (discussed below).

Once the inputs and outputs are defined, the processing methods to transform the inputs into the outputs can be specified. This step will be the basis for determining if conventional programming or Expert System paradigms can be used. This determination will help guide the way the collected data can be organized and stored in the various compartments of the Knowledge Base.

Collect Data Related to Feedwater Heaters

During this step, the data related to the operation, maintenance, and performance of feedwater heaters can be collected and categorized. The data sources can be divided into two categories: generic and plant specific. Generic sources include industry-sponsored and in-house work at Babcock & Wilcox. Some current generic data sources are: Symposium on State-of-the-Art Feedwater Heater Technology (EPRI CS/NP-3743); Recommended Guidelines for the Operation and Maintenance of Feedwater Heaters (EPRI CS-3239); Nuclear Plant Feedwater Heater Handbook (EPRI NP-4057); Life Extension for Closed Feedwater Heaters (ASME Pressure Vessel and Piping Conference, June 1985); Standards for Closed Feedwater Heaters (Heat Exchange Institute, August 1984); Secondary Plant Performance Monitoring Evaluation, Surry Power Station Units 1 and 2 (Babcock & Wilcox, April 1986), and Feedwater Heater Systems (Central Electricity Generating Board, U.K. 1971).

Plant-specific data can be found in the performance, maintenance, and operations areas. Specific utility maintenance and performance data can be included in the

Knowledge Base. These data will include, for example, the following performance data: shell-side pressure drop, tube-side pressure drop, inlet and outlet temperatures (TTDs & DCAs), and flow rates. In addition, maintenance data such as tubesheet diagrams, location and date of plugged tubes, type of failure, failure cause or conjecture, and mode of operation at the time of failure will be collected.

The collected data can be reviewed for relevance, obsolescence, and consolidation. The data can be sorted and indexed effectively to match the user requirements discussed above. The collected data can also be reviewed with respect to organization of and placement in the Knowledge Base. The data can be organized in a systematic form such that a computerized database management system (DBMS) can be selected.

Select Hardware/Software Deployment Platforms

The selection of both hardware and software shall address both functional and commercial aspects. This step addresses all appropriate aspects in this area and defines target development and deployment platforms for both hardware and software.

Regarding hardware, the ideal product would be deployable simply as a portable program (with documentation) that could be installed on as large a range of typically configured PCs as possible. This would enhance the user attractiveness by not requiring uncommon PC hardware configurations and/or additional specialized hardware purchases where PCs are already integral to the user environment. This capability would preempt all the potential negative pitfalls associated with trying to market a software product, but being constrained to specific hardware features.

Initially, targeting the hardware deployment platform as an IBM (or compatible) PC/AT with 640K memory, a hard disk of at least 10K, and possibly an enhanced graphics adaptor (EGA) and monitor would be reasonable. In today's environment, this configuration represents a fairly common platform and would minimize user resistance from this perspective. The functionality of the product may drive the configuration upwards in size and complexity. Starting from this point, however, can ensure that the product development occurs with pressure to minimize the required hardware configuration to enhance user acceptance, as opposed to uncontrolled functional development with no consideration of the intended user hardware environment.

Regarding software, using a credibly commercial traditional database shell and an integrated credibly commercial "expert system" shell as the basic structure of the product would be appropriate. This can provide the advantages of commercially supported software for aspects of program functionality (e.g., database manipulation utilities) not peculiar to the specific application, while also freeing project resources for development of the application specific aspects.

One example candidate for the database shell is Expert-EASE's EASE+ database and interface package. This package is specifically designed for engineering database applications. Its interactive graphics interface, coupled with many functional utilities associated with typical manipulations of engineering data (e.g., trending and plotting), make it attractive for engineering applications.

One example candidate for the "expert system" shell is Neuron Data's NEXPERT. This shell provides a "production rule" environment, one of the specific Artificial Intelligence (AI) paradigms that has been successfully implemented in many practical applications. More importantly, this shell also provides an "object-oriented" environment, the other AI paradigm that has found practical application. This object-oriented environment is not found in many PC-based expert system shells, yet

can be an extremely important mechanism for data (knowledge) representation in certain cases (e.g., hierarchical organized data/objects).

The preceding discussion provides a general perspective, illustrated with specific examples, of the approach that can be used to define the hardware and software deployment platform for this project. Some specific factors that will be addressed will include:

1. State-of-the-art in commercially deployed PC hardware is changing rapidly, e.g., introduction of 386 processors, larger available memory and disk size environments, new operating systems, etc. Care should be taken to use current, but not risky, advances in available technology.

2. Networking requirements, as appropriate (e.g., access to a plant database resident in another machine), should be assessed.

3. Commercial viability, as well as product functionality, for shell package vendors (e.g., Expert-EASE Systems, Neuron Data) should be assessed. Items such as established user base, user support experience, and product enhancement history are some key evaluation criteria.

These and other appropriate issues and aspects of PC technology and the target user environment should be analyzed for both functional capability and user acceptance.

<u>Define Knowledge/Database Structure Requirements</u>

With the user's needs and the input requirements for each feedwater heater activity defined, functional requirements specifications, a detailed definition of the data characteristics, can be developed. This includes such items as the data type, field length or precision. The data models for each of the applications should provide the information necessary for the physical design of the database.

Selecting a conventional database management systems (DBMS) should depend on the database structures best suited to support an LCA activities. A feedwater heater LCA can be built under an applications shell that can either provide the required DBMS functions or be capable of interfacing with a variety of commercially available systems or both.

Much of the compiled data/knowledge base will be information readily stored with conventional database technology. Numerical data, which must be subsequently numerically analyzed, is a typical example. For a feedwater heater, this might include performance data such as temperatures, pressures, flows (if available), and levels. Typically, these data are routinely logged for subsequent analysis and/or trending. Other examples might be instrumentation calibration records, scheduling plans, and maintenance records. A definition of what types of identified data are best encoded in a traditional database structure can be determined, and a specification for the selection of the DBMS can be developed.

A substantial portion of the data/knowledge base, however, is anticipated to be compiled with AI-based structures, specifically associated with production-rule and object-oriented paradigms. The codification of the knowledge itself into the knowledge base should be closely coordinated with its associated processing mechanism.

Production rule format would be most applicable to procedural type knowledge, especially in areas where heuristic and/or non-deterministic decision making is employed. However, not all procedural knowledge will be implemented in this manner,

so procedure codification should be coordinated with the processing mechanism as a function of both the nature of the knowledge and how the system will best process it.

Object-oriented processing is the other commercially robust AI-based processing mechanism, and is most applicable to representing objects, whether conceptual or physical, with hierarchical or other types of relationships. It is anticipated that most of the knowledge base, including, for instance, sets of associated production rules themselves and possibly sets of numerical data, will be encoded in this format. This encoding will allow the subsequent use of standard AI-based object-oriented processing mechanisms to be used for the processing as appropriate.

As alluded to earlier, while some procedures are anticipated to be encoded in the production rule format, other procedures, directly aligned with specific objects and specific initiating event circumstances, would be best encoded as "methods" tied to "slots" (i.e., attributes) associated with the related object. In many cases, these methods would be initiated as demanded (demand-driven), i.e., the procedures would be activated when their associated outcome was triggered by the main analysis processing mechanism. One use for this technique is consistency checking. If the value of an attribute of an object in the Knowledge Base is modified, methods can be automatically triggered to check associated data for inconsistencies. For example, if a change in level setpoint is recommended, consistency check methods should be automatically initiated (and they would be using this technique) to check for recommending a parallel change in high- and low-level alarm and trip setpoints.

Define Analysis/Processing Requirements

As discussed above, the structure for encoding the appropriate data/knowledge is integrally related to the associated analysis/processing techniques. This step addresses the definition of what programming paradigms, conventional or AI-derived, are appropriate for the activities and associated data/knowledge bases implemented in an FHLCA.

Certain numerical data, such as performance data, may be best encoded in a traditional database structure. Given stored data such as pressures, and temperatures, related performance parameters, such as terminal temperature difference and drain cooler approach temperatures, can be calculated and trended. Alarm limits can be set and excursions can initiate diagnostic and/or recovery procedures. Possibly even reliability checking of raw input data would be appropriate, if not already accomplished, e.g., by the plant process computer, if this were the data source. All these activities are fairly algorithmic in nature, and can be accomplished with conventional techniques. If so, the capabilities of the commercial database shell can be readily used in processing and displaying such analyses.

When an analytical operation is neither possible nor feasible with a database shell, a high level language (C) can be used. These operations should be identified in conjunction with any particular programming requirements. The data analyses that are best encoded with traditional database analysis techniques and conventional programming approaches can be identified.

A substantial portion of the Knowledge Base, as previously discussed, is anticipated to be compiled with AI-based structures, specifically targeted to be processed with AI-derived production-rule and object-oriented paradigms.

Production rule processing would be most applicable to procedural-type knowledge, especially in areas where heuristic and/or non-deterministic decision making is

appropriate. In cases where the analysis is data-driven (e.g., given certain data, draw appropriate conclusions and/or make appropriate recommendations), the forward-chaining processing mechanism would be employed. A possible example is examining all the ramifications of a specific operational or maintenance activity.

In cases where the analysis is goal-driven (e.g., how can a specific conclusion or recommendation be drawn or justified), the backward- chaining processing mechanism would be employed. An example is querying the system to determine if the feedwater heater is operating normally, when off-normal valve lineups may affect analysis results predicated on normal conditions. Another example might be requesting the system's advice on when to replace a tube bundle. In this case, the backward-chaining mechanism would allow the system to search through various different types of data/knowledge (e.g., current status of tube bundle, degradation trends, already planned and scheduled outages, lead time on required repair materials, estimated repair time itself, special equipment requirements and availability) to determine one or more scenarios where the goal is met (a recommended repair schedule) with all required constraints satisfied. A further advantage of this approach could be identifying alternative scenarios, even some where all constraints are not met, for comparative trade-off. In all cases, this processing mechanism can "explain" its reasoning path by relating its chained reasoning path, thus allowing the user to understand how the conclusion was reached.

Both these chaining mechanisms can be used as appropriate in the various analysis stages. This processing mechanism presumes that the procedures, non-deterministic or otherwise, have been encoded in the production rule format. Not all procedural knowledge will be implemented in this manner, so the procedure codification will be coordinated with the processing mechanism as a function of both the nature of the knowledge and how the system will best process it.

Object-oriented processing is the other commercially robust AI-based processing mechanism, and is most applicable to codifying objects, whether conceptual or physical, with hierarchical relationships. As noted earlier, it is anticipated that much of the data/knowledge base, including, for instance, sets of associated production rules themselves and possibly sets of numerical data, could be encoded in this format. Standard AI-based object-oriented processing mechanisms will be used as appropriate for the analysis involved.

One example, discussed earlier, was automatic consistency checking using the object-oriented paradigm. Another example might be searching through an allowable materials list and consumable materials control program to determine consistent sets of procedures and materials for a tube bundle, tubesheet, or shell repair. As with the production rule processing technique, this example presumes that the materials and procedures information have been encoded within the object-oriented structure of parts and attributes. In addition, several alternatives can be sought and ranked by predefined optimization schemes (e.g., earliest time, minimal cost).

In all cases, the codification of the knowledge itself into the knowledge base must be closely coordinated with its associated processing mechanism.

Define Interactive User Interface Requirements

An FHLCA will provide a wide range of functions for a diverse group of users including personnel from operations, maintenance, engineering, and management. Each group has different needs, requiring different data to be extracted from the Knowledge Base, requiring the information to be processed in diverse ways, and requiring that the processed output be examined in a familiar and comfortable manner. To satisfy these requirements, it is necessary to design both a "standard"

and a "smart" interactive user interface that is flexible, responsive, and easy to operate.

The "standard" interface can take the form of a hierarchical menu structure and should accept input by one or more methods, e.g. keyboard entry, cursor positioning, mouse input. This interactive interface can allow the user easy access to those menus, as well as the ability to abort any operation. The interface can also provide access to the appropriate output functions, such as tabular or graphical displays.

The "smart" interface may actually be multiple interfaces as a function of the type of user. This interface can, acting through the "standard" interface, direct the user to provide more information where incomplete or inadequate, perform some preliminary calculations, or determine which analytical method(s) to use. This interface, making use of Expert System techniques where appropriate, can aid the user in defining the problem to be approached (i.e., input requirements and method of solution), and, perhaps, suggest specific output forms and/or additional related problems to solve.

One aspect of a user interface is data security. This will be provided through password access levels. In its simplest form, each password level provides access to a specific set of menu items. Considering a system with multiple levels of password protection, the extremes are:

- o All persons with the lowest password level will have access to only the training menus. They will be able to view or print but not change any of the data.
- o All persons with highest password level will have access to all menu options, and accordingly be able to add information to the database, modify existing data, and perform system maintenance.

Product (FHLCA) Development

With the Knowledge Base structure defined, the identified data/knowledge can be entered. Since development can proceed using the rapid prototyping strategy, the structure definition and data input can occur in a series of steps. This can provide the development team the opportunity to use the Knowledge Base, identify missing data, discard non-essential data, reclassify data deposition, use of expert system paradigms, interface with the user, and interface with the analysis tools in small manageable pieces (versus trying to fill the entire database, and then discovering a problem!). The feedback and experience gained at each step will allow for more efficient execution of future steps and provide the opportunity to catch and correct errors/problems before they become inculcated in the software package and greatly impact cost and schedule.

The menu structure and the specific options, determination of how/if windows will be used and their implementation, use of graphics, etc. can be coded based on the functional requirements. Since the menu structure definition will occur in a series of steps, the development team has the opportunity to develop menus and other interfaces as needed. Care must be used to ensure that the final user interface provides the user with a coherent, fluid tool (rather than a patchwork of interfaces). The feedback and experience gained at each step will allow for redefinition of menu items and use of different interfaces.

For the processing of information from a traditional database), a high-level language (C) can be used. For production rules and/or object-oriented programming, an expert shell (such as NEXPERT) will be used.

Since development will proceed using the rapid prototyping strategy, the writing (and testing) of the analytical processing software will occur in a series of steps. This will provide the development team the opportunity to design and use various interfaces with the database, use the expert system shell, identify missing data, discard non-essential data, reclassify data deposition, interface with the user, and interface with the analysis tools in small manageable pieces (e.g., trying to fill the entire database, and then discovering a problem!). The feedback and experience gained at each step will allow for more efficient programming, use of expert system shells, and use of the interfaces.

Software can be developed and tested during each "slice" of the rapid prototyping process to ensure accurate translation of the requirement specifications and correctness of the coding. Test cases can be recorded and saved, and repeated after the final product is complete to ensure successful integration of all program parts.

While the process of rapid prototyping will facilitate the preparation of the software documentation, a large portion will be written after a prototype FHLCA has been developed. With respect to commercially purchased software programs, no attempt will be made to summarize or condense the provided documentation; however, a Programmer's Guide should include any information required for successful interfacing and operation that it not provided by the supplier.

CONCLUSIONS

This paper presents the steps necessary for the development of an FHLCA making use of currently available computer tools, such as graphics, database management systems, available expert system shells on PC. The steps involved the collection of information and knowledge, determining appropriate hardware and software, determining requirements for knowledge/database structures, traditional and AI-derived "analytical engines" and user interfaces, and, finally the "filling" of the Knowledge Base, and writing of the code, testing, and documentation.

The final package will assemble the knowledge of experts, as well as information contained in various reports, into a software tool that can easily be used by a variety of utility personnel to assist them in making decisions concerning feedwater heaters. As new data become available, the data/knowledge base should be updated to reflect those changes.

APPLICATIONS

Technical Specifications Advisor Pilot Project for Brunswick Steam Electric Plant—Unit 1

S. A. LAUR
Carolina Power & Light Co.
Corporate Nuclear Safety Section
P.O. Box 1551
Raleigh, North Carolina 27602, USA

ABSTRACT

The purpose of this project was to determine the feasibility and acceptability of a "Technical Specification Advisor" expert system. Envisioned was a knowledge base containing the interrelationships among operable/inoperable Technical Specification equipment and the limiting conditions for operation (LCO's) for each mode of reactor operation. The Technical Specification Advisor functions on the same level as the Shift Foreman.

The commercial program M.1 was initially utilized for the "Technical Specification Advisor" pilot program. The expert system pilot project has been successfully converted to Prolog. Compiled BASIC was utilized to provide an interface between the user and the expert system.

This project demonstrated the feasibility of a personal computer based Technical Specification Advisor for a small subset of Technical Specifications for one mode of operation. Expansion of the scope and potential enhancements are also considered in the report.

INTRODUCTION/OVERVIEW

The Brunswick Steam Electric Plant is comprised of two General Electric Boiling Water Reactor nuclear units located at Southport, North Carolina. The Technical Specifications describe certain minimum equipment which must be operable for each allowed mode of reactor operation. These Technical Specifications ("Tech Specs") delineate certain "Limiting Conditions for Operation" (LCO's) and specify actions to be taken for each safety-related system which is inoperable for some reason.

The Tech Specs are a regulatory as well as a technical document. As such, they contain a combination of legal-ease and technical jargon which makes them difficult for the untrained person to comprehend. Additionally, there are examples where several specifications within the Tech Specs are applicable for a given piece of equipment, and other cases where the inoperability of one system affects the operability of others. A knowledge of the various individual subjects

alone is not enough for competent compliance--one must additionally be aware of the various cross-references and modifiers within the document as a whole.

The Shift Foreman is a trained, NRC licensed Senior Reactor Operator who is in charge of the shift. He is, among other things, the expert concerning the Technical Specifications for the unit. When maintenance is required for a piece of equipment, a component failure occurs, or a system is found to be out of specification during surveillance testing, the Shift Foreman ensures compliance with the Tech Specs (as well as a myriad of other procedures) by means of his training, knowledge, and experience.

The purpose of this project was to determine the feasibility and acceptability of a "Technical Specification Advisor" expert system. The scope of this pilot project was large enough to demonstrate the feasibility and acceptability of using a knowledge engineering approach to Tech Spec compliance but at the same time was small enough to be performed with limited corporate resources.

This report addresses the reasons for pursuing an automated Technical Specification system and the application of expert system technology to this problem. It then describes the Technical Specification Advisor Pilot Program. Finally, future opportunities for application and enhancement are considered.

STATEMENT OF NEED/PROJECT REQUIREMENTS

The Technical Specification Advisor Pilot Project was begun in order to fulfill a perceived need. The scope of the project was selected to allow evaluation of an expert system approach to fulfill this need without great expenditure of resources. Several goals were identified for the project. The perceived need, scope, and project goals are discussed below.

Project Desirability

The Technical Specifications at a nuclear power plant represent a complex regulatory and technical document. Even if the wording were made completely impervious to differences in interpretation, the interrelationship among various specifications in that document complicates the determination of their applicability in any given condition. Even experienced and well-trained operators occasionally fail to properly apply the "Action Statements" required by certain "Limiting Conditions for Operation" (LCO's).

One sort of error the Shift Foreman may make involves failure to recognize that a system or component is inoperable. The word "OPERABLE" is a defined term in the Tech Specs; it is defined in the Brunswick Tech Specs as follows:

> OPERABLE - OPERABILITY A system, subsystem, train, component, or device shall be OPERABLE or have OPERABILITY when it is capable of performing its specified function(s). Implicit in this definition shall be the assumption that all necessary attendant instrumentation, controls, normal and emergency electric power sources, cooling or seal water, lubrication or other auxiliary

equipment that are required for the system, subsystem, train, component, or device to perform its function(s) are also capable of performing their related support function(s).

This is a rather broad definition, and results in necessary Tech Spec actions for inoperable systems which may not be covered in the Tech Specs directly.

Another sort of error is failure to apply all applicable Tech Spec action statements. Often a Limiting Condition for Operation will have one set of actions for inoperability of a given system or component, and another set if related systems are simultaneously inoperable. Plant status at the time a system or component becomes inoperable is, therefore, important. Examples include the high pressure coolant injection specification, which ties in the operability status of the automatic depressurization system and low pressure core cooling systems.

The final class of error considered here involves failure to interpret the meaning or applicability of the Technical Specifications in the same way that the NRC does. This sort of error may occur even when both parties endeavor to meet the spirit and letter of the specifications.

These failures to adequately comply with the Technical Specifications incorporated into the license result in the submittal to the NRC of a Licensee Event Report (LER). Depending upon the nature of the non-compliance, NRC may issue a Notice of Violation. Depending upon the severity of the violation and other factors, the result may be a civil penalty (fine).

An automated Technical Specification Advisor could eliminate some of the non-compliances presently experienced. The system described in this report would <u>not</u> help in the case of failure to recognize that a system was inoperable; however, it <u>would</u> assure that the operators were made aware of all applicable LCO's for the given plant condition. It could additionally function to store the results of clarifications or interpretations which arise out of differences in understanding between the utility and NRC. The potential benefits of the Technical Specification Advisor include fewer LER's, violations, and fines.

<u>Project Scope</u>

The purpose of this project was to determine the feasibility and acceptability of a "Technical Specification Advisor" expert system. Envisioned was a knowledge base containing the interrelationships among operable/inoperable Technical Specification equipment and the Limiting Conditions for Operation (LCO's) for each mode of reactor operation. The "expert" being emulated by this system is the Shift Foreman on watch in the control room of a nuclear power station. It is assumed that the status of Technical Specification systems, that is, "operable" or "inoperable," is determined separately and input into the Technical Specification Advisor. The system utilizes this data and the rules developed from the Technical Specification "Limiting Conditions for Operation" (LCO's) to arrive at what action is required in order to be in compliance with the specifications.

The scope of the pilot program was large enough to demonstrate the application of a knowledge engineering approach to Technical Specification compliance for a small subset of Technical Specifications. Technical Specifications involving only the following areas were considered for the Technical Specification Advisor Pilot Project: Mode of reactor operation was limited to Operational Condition 1 (Power Operation) and equipment was limited to the Emergency Core Cooling Systems (ECCS), the Standby Liquid Control system (SLC) and the 4160 volt AC electrical power specification (3.8.1.1). The general LCO caveats contained in Tech Spec section 3.0 were also included (Limiting Conditions 3.0.3 and 3.0.5) to provide demonstration of the interplay between multiple sections of the Tech Specs.

With these restrictions, the project became manageable without a great expenditure of resources. The inclusion of the general LCO requirements of section 3.0 tested the ability of the system to function properly in an area which has resulted in occasional non-compliance in the past. The rule base developed to handle these LCO's consisted of approximately ninety rules.

Project Goals

As stated above, the Technical Specification Advisor Pilot Project was developed to demonstrate the feasibility and acceptability of utilizing expert system techniques to aid in Technical Specification compliance. Feasibility is a multifaceted criterion. For example, it relates to the ability of the system to duplicate (or surpass) the Shift Foreman's ability to comply with the complicated, intertwined rules within the Technical Specifications. It also involves the cost-effectiveness of the proposed system. By "acceptability" is meant not only the acceptance of the system by the ultimate end-user on shift, but also the justification of the project to corporate management.

One feasibility goal of the project, then, was the demonstration that the system could properly determine the correct LCO's and proper actions for any set of plant conditions within the scope of the model. A second goal was to determine whether this system could be effectively implemented on a personal computer or similar low cost platform. Third, it was hoped that acceptance of the basic idea by operators at the Brunswick site would be obtained. Finally, the pilot program would provide substantive evidence of the soundness of the concepts and serve to facilitate approval by company management of a larger, full-scale model.

APPLICATION OF EXPERT SYSTEM TECHNOLOGY/METHODOLOGY

The purpose of the "Technical Specification Advisor" expert system is to emulate the Shift Foreman on watch in the control room of an operating nuclear power plant. He is the person charged with compliance with the Technical Specifications, and it is his expert knowledge which assures such compliance even given the interrelationships discussed above. The goal is for the expert system to reach the same, proper conclusions as to what action is required as the Shift Foreman. The expert system provides the added benefit of not forgetting to consider related system operability status when

making the determination as to what Technical Specification actions are required.

It should be pointed out that there are other "experts" associated with Technical Specification issues which could be emulated by an expert system, but which are not a part of the scope of the "Technical Specification Advisor" expert system. For example, in many cases an engineering evaluation must be performed to determine whether or not an "as found" condition affects the operability of a system. Site management frequently makes interpretations of obscurely-worded or unclear specifications. Both of these cases involve expert opinions which conceivably could be captured in an expert system. However, the "Technical Specification Advisor" starts at the point of "operability" and "inoperability" for the applicable systems. From this information the expert system determines, just as does the Shift Foreman, what must be done to comply with Technical Specifications.

The incorporation of a subset of the Brunswick Tech Specs into a knowledge system involved selection of a suitable problem domain, knowledge acquisition, software selection, and development of the knowledge base itself. The selection of a suitable problem domain was discussed above. The knowledge acquisition, software selection, and knowledge base development methodology are discussed in the following paragraphs.

Knowledge Acquisition

The most readily available and obvious source of raw data for this project was the Brunswick Steam Electric Plant, Unit 1 Technical Specifications. The specifications from the "Limiting Conditions for Operation" section of that document, as narrowed down by the reduced scope of this project, served as the initial information for the Tech Spec Advisor. However, the nature of the Technical Specifications, since they constitute a regulatory as well as technical document, is such that a knowledge of the various subjects alone is not enough for competent compliance--one must additionally be aware of the various cross-references and modifiers within the document as a whole.

For this reason a human expert, the Shift Foreman, is relied upon to ensure that the Tech Specs are complied with during operation. The knowledge engineer developing the Technical Specification Advisor required a similar expert in the field of Brunswick Unit 1 Technical Specifications in order to develop the knowledge base, design the user interface, and validate the resulting system. A person knowledgeable in Standard Technical Specifications was available in the General Office for the initial stages of the project. However, in the course of validating the expert system, it became obvious that some of the plant-specific knowledge, such as interpretations and regulatory decisions affecting Technical Specifications, required a higher level of expertise. This was obtained from a Brunswick Site Regulatory Compliance engineer holding an NRC Senior Reactor Operator's license on the Brunswick units.

Software Selection

Basically, the Tech Specs as written consist of numerous rules which are readily adapted to an "if-then" format. The LCO's represent a

"goal" sought by the Shift Foreman whenever the status of Tech Spec equipment changes. The applicable LCO's dictate what action must be taken as a result of a change in equipment status. In many cases, more than one LCO will apply at a time. In other words, LCO is a multi-valued goal for the Shift Foreman to pursue; the failure to locate all applicable LCO's in a given circumstance may result in the plant being operated outside of the license.

In addition to finding the applicable LCO's it is important for the Shift Foreman to understand the reasons for his selected action. Often, the result of complying with the action statement of an LCO is the orderly shutdown of the unit. This represents a large economic cost to the utility in terms of lost electric power generation. On the other hand, failure to recognize applicability of a given LCO may result in the violation of regulations or the plant operating in an unsafe condition. The Shift Foreman therefore must be able to explain and justify the applicability (or inapplicability) of the LCO and the actions he has taken as a result.

For the above reasons, expert system software was sought which supported "if-then" rule format and was "goal-driven." (Another way of expressing goal-driven is "backward-chaining.") An explanation facility was also desired. Reasoning with incomplete data, confidence factors, or "fuzzy logic" was <u>not</u> a requirement for this project. Finally, software was chosen based upon availability, cost, compatibility with existing personal computers, and ease of use for first-time programmers.

M.1, an expert system "shell" marketed by TeKnowledge Corporation, was used for the initial version of the pilot program. Prolog, an artificial intelligence language, was used for a second version of this expert system. There are advantages and disadvantages corresponding to each choice.

The commercial program M.1 was initially chosen for several reasons. Basically, the program employs a goal-driven, backward chaining inference engine and readily supports the "if-then" rule format. It also allows the use of multi-valued goals, so that it can be made to search for <u>all</u> values of a parameter (LCO, for example) which match the existing world state. The software also keeps track of the justifications for its conclusions (in the cache memory), which facilitates providing explanations for the program's results.

The Technical Specification Advisor expert system was easily converted to Prolog, a popular artificial intelligence language. The version utilized was ED Prolog from Automata Design Associates. Being a language, rather than a "shell," Prolog provided more flexibility than M.1 in some ways; however, it required more programming expertise. There is a learning curve associated with either product, however.

A sample rule in each system demonstrates the syntax and facilitates discussion of some of the salient features. In the following rules, "css_a" represents the "A" train of the Core Spray System; "css_b" is the "B" train. Also, "op" refers to "operable" and "css" refers to the entire Core Spray System:

```
In M.1:    '3.5.3.1-3':if css_a = op
                       and css_b = op
                       then css = op.
```

```
In Prolog: css(op) :- css_a(op), css_b(op).
```

In the M.1 version of the rule, the information up to the colon is a label for the rule. The rule itself almost reads like English: "If Core Spray train 'A' is OPERABLE and Core Spray train 'B' is OPERABLE then the Core Spray System is OPERABLE."

The Prolog rule is a little less straightforward, but also simpler. The symbol ":-" is read "if" and the comma indicates a logical "and" connective. In English: "The Core Spray System is OPERABLE <u>if</u> Core Spray train 'A' is OPERABLE and Core Spray train 'B' is OPERABLE." The rules say the same thing; the Prolog version just puts the conclusion first.

M.1 has a built-in explanation facility. If one were to seek the goal "css" in a system containing just the one rule above, M.1 would ask the user for the value of css_a and css_b. If both had the value "op," M.1 would conclude that css had the value "op" also. It would state by way of explanation: "because 3.1.5.1-3." In other words, M.1 gives the <u>label</u> of the rule which allowed it to reach a conclusion as the reason for reaching that conclusion. This explanation may or may not be meaningful, depending upon whether the rules in the knowledge base can be labeled in a manner which would provide useful data as to why M.1 reached a given conclusion.

Prolog, being a language, is more general. It has no built-in explanation facility, so this must be programmed in. In the Prolog rule, the term "css" has an argument, "op" (for operable) in parentheses. The number of arguments is not limited in Prolog, so that the explanation facility can be easily added by utilization of additional arguments. In the modified Prolog rule below, the string "Explanation" is a variable (it begins with an upper case letter):

```
css(op,Explanation) :- css_a(op), css_b(op),
     Explanation = 'Both Train A and B CSS are OPERABLE'.
```

If the above rule is true, that is, if css_a(op) and css_b(op) are factual, Prolog will assign the sentence delimited by the apostrophes to the variable "Explanation." Similar methodology was utilized in the Prolog version of the Technical Specification Advisor to build up multiple explanations. Such explanations have the advantage of being more specific and detailed than the rule label supplied by the built-in M.1 explanation facility.

<u>Methodology</u>

Expert systems often inherently allow rapid prototyping, system testing, and subsequent model enhancement as an iterative development methodology. This is true because the knowledge base is not sequentially accessed but rather triggered as a result of the existing world picture at each point. Rapid prototyping, system tryout, and model enhancement for the Technical Specification Advisor were facilitated by the choice of project scope, software, and verification methodology.

The pilot project utilized a limited scope, allowing a short amount of time between system conceptualization and a working model. This

scope, as discussed previously, was chosen to demonstrate key, desirable system features without a great expenditure of programming time and effort. The selected Technical Specification LCO's and corresponding action statements were entered into the knowledge base as facts and rules. The simulated state of the plant condition (ie, operable or inoperable) was contained in a data file. The expert system utilized the rules within the knowledge base and the facts of known plant conditions to seek applicable LCO's.

A number of permutations of plant conditions were utilized to try out each step in the model, and corrections were made to the knowledge base as necessary to cause the model to behave as desired. Upon completion of the small model, various combinations of plant conditions were simulated and the outputs of the model recorded.

When the model had been refined to the satisfaction of the knowledge engineer, an "expert" from the plant was utilized for further validation. This expert was an NRC-licensed Senior Reactor Operator at the Brunswick site, who additionally possessed a great deal of experience in the area of Technical Specification compliance issues. This expert posed various plant configurations, within the scope of the model, and commented upon the validity of the LCO's chosen and the action statements generated by the system. Any discrepancies noted were then factored into the knowledge base.

Basically, then, the methodology was an iterative one. The knowledge engineer utilized the base document, Technical Specifications, to enter the various rules and facts. Simulated configurations were utilized to verify the working of key points of the model and point out required corrections. The system was next compared with a true expert on the system, and the knowledge engineer then made the necessary adjustments. The process then continued the verification cycle.

It should be noted that expansion of the knowledge base to cover additional LCO's is, in general, relatively straightforward. This is because a rule-based, goal-driven system does not act in a serial manner as do traditional programming languages. The same iterative methodology may be used throughout to build a complete model of the Technical Specifications, adding in a few rules at a time and testing.

DESCRIPTION: THE TECHNICAL SPECIFICATION ADVISOR

Having considered the desirability of utilizing expert system technology to solve the identified need, the Technical Specification Advisor Pilot Project was begun. This section discusses the prototype expert system which was developed. First, the assumptions made in developing the model are presented. Next, the model itself is described, including implementation of LCO's into M.1 and Prolog rules. The user interface developed for the project is considered next. Finally, the methodology used to validate the expert system model is discussed.

Assumptions and Initial Conditions

The scope of this pilot project was limited as discussed above in order to provide a readily developed prototype small enough to be

performed with limited corporate resources. Enough complexity had to be included, however, in order to demonstrate the feasibility and acceptability of using a knowledge engineering approach to Tech Spec compliance.

In addition to narrowing the scope of this pilot project to a subset of the Tech Specs, some simplifying assumptions concerning the equipment have been made, particularly concerning emergency diesel generator AC electrical power supplies for some equipment. The Brunswick site has two nuclear units, and only the Tech Specs for Unit 1 are modeled here. This poses no problem in most instances, as each unit has a full complement of required safety-related equipment. In the case of diesel generators, however, instead of having two dedicated emergency diesel generators per unit, the designers of the plant decided to provide maximum flexibility by providing four, interconnected diesels. By interconnected is meant that a given diesel supplies power to loads for both units, not that the electrical buses themselves are connected.

For this reason, specification 3.0.5 leads to a far more complicated result than if the four diesels were arranged two exclusively to each plant. This is also why the Unit 1 Tech Specs (and the Unit 2 Specs also) require four diesels to be operable in specification 3.8.1.1. Sufficient interplay between the various specifications still exists if each site is considered as having its own pair of emergency diesel generators.

Therefore, in the pilot project, it was assumed that Brunswick Unit 1 has two safety related, redundant trains of equipment, A train and B train. Each train is assumed to receive power from either its normal AC electrical bus or from its respective emergency diesel generator. No cross connection between units is assumed. This can be readily corrected when the project is expanded.

Although there are a great number of safety-related loads powered off of these electrical buses, this model considers only the Core Spray System (CSS), Low Pressure Coolant Injection (LPCI) system, and Standby Liquid Control (SLC) system. In addition, the following non-redundant or non- AC powered equipment is included in the model: The High Pressure Coolant Injection (HPCI) System, the Automatic Depressurization System (ADS), and the LPCI train A to train B cross connect valve.

Model Description

The Technical Specification Advisor pilot project was initially developed using the expert system shell, M.1. It has now been successfully converted to Prolog. Both M.1 and the version of Prolog utilized run on IBM or compatible personal computers. Each has its relative strengths and weaknesses, but Prolog appears to be the more general choice given that there is a learning curve for using either product.

The Technical Specification Advisor is a goal-driven, backward-chaining, rule-based system. This type of system is well-suited to modeling the Technical Specifications. The top-level goal is "LCO," and the system seeks to prove the truth (or applicability) of the given LCO by determining whether the conditions of plant equipment support the truth of the LCO. In such a case, the system outputs the

action statements applicable to the LCO which has been proven. In the case of the Technical Specification Advisor, the expert system is designed to find all examples of LCO's which are applicable based upon the plant conditions and output all the appropriate required actions.

One desirable feature of a diagnostic expert system, which in a sense describes the Technical Specification Advisor, is the ability of the system to explain not only what it has concluded, but why. This feature was better implemented within the Prolog version of the Technical Specification Advisor, so that the operator not only receives a listing of all applicable LCO's and related action, but also what equipment status led to each one. In this way the operator can assign relative importance to the maintenance and repair of inoperable equipment and so minimize the impact on plant operation.

Description of Rules

Once the scope of the pilot project was defined and knowledge acquisition begun, the information was incorporated into the knowledge base in "if-then" rule format. A discussion of some of the salient points of the Technical Specification Advisor knowledge base follows.

An example serves to demonstrate the relationship between a typical Technical Specification LCO and the corresponding rule or fact in the knowledge base. The following is a portion of LCO 3.5.1:

> LIMITING CONDITION FOR OPERATION
>
> 3.5.1 The High Pressure Coolant Injection (HPCI) system shall be OPERABLE with:
> a. One OPERABLE high pressure coolant injection pump, and
> b. An OPERABLE flow path capable of taking suction from the suppression pool and transferring water to the pressure vessel.
>
> Action:
> a. With the HPCI system inoperable, POWER OPERATION may continue provided the ADS, CSS, and LPCI systems are OPERABLE; restore the inoperable HPCI system to OPERABLE status within 14 days or be in at least HOT SHUTDOWN within the next 12 hours and in COLD SHUTDOWN within the following 24 hours.
> b. . . .

The rule in M.1 which implements part of this LCO is:

```
'3.5.1-1':if   hpci = inop and ads = op
               and css = op and lpci = op
          then lco = '3.5.1: Restore HPCI within 14 days or
               be in hot shutdown within the next 12 hours
               and cold shutdown in the following 24
               hours.'.
```

This reads just like English. The corresponding rule in Prolog is:

```
lco('3.5.1',yes,Exp)  :- hpci(inop), ads(op), css(op),
                         lpci(op), Exp = ['HPCI is
                         inoperable'].
```

This may be read:

> "LCO 3.5.1 has a value 'yes' for the reason 'Exp' _if_ HPCI is inoperable _and_ ADS is operable _and_ Core Spray System is operable _and_ LPCI is operable; if such is the case, assign the list 'HPCI is inoperable' to the variable 'Exp'."

Note that the M.1 explanation facility will give the reason "because 3.5.1-1;" the Prolog version has been programmed to remember that HPCI being inoperable is the reason this rule fired.

The two specifications in the "applicability" section of Tech Specs, 3.0.3 and 3.0.5, warrant some explanation here. Basically, the Limiting Conditions for Operation (LCO's) define the required equipment configuration for each allowed mode of reactor operation and specify what action to take in the event that a degraded condition arises. Action statements specify not only what degraded states are allowed but also the allowed outage time for the inoperable equipment.

Specification 3.0.3 applies generally to all LCO's and states:

> In the event a Limiting Condition for Operation and/or associated ACTION requirements cannot be satisfied because of circumstances in excess of those addressed in the specification, the unit shall be placed in at least HOT SHUTDOWN within 6 hours and in COLD SHUTDOWN within the following 30 hours unless corrective measures are completed that permit operation under the permissible ACTION statements for the specified time interval as measured from the initial discovery or until the reactor is placed in an OPERATIONAL CONDITION in which the specification is not applicable.

This means, basically, that if the LCO cannot be met, either directly or through reliance on an action statement, then the plant must be shut down. In other words, 3.0.3 is the "catch all" action statement for subsequent LCO's.

One of the rules associated with specification 3.5.1 handles the 3.0.3 case as follows (M.1):

```
'3.5.1-2':if   hpci = inop and
               not(ads = op and css = op and lpci = op)
          then
               lco = '3.0.3: Place the unit in Hot Shutdown
               within 6 hours and cold shutdown within the
               following 30 hours.'.
```

And in Prolog (with the semi-colon denoting a logical "or" connective):

```
lco('3.0.3',yes,Exp)  :- hpci(inop), lowpressure(inop,Exp1),
                         append(['HPCI is inoperable']
                         ,Exp1,Exp).

lowpressure(inop,Exp) :- ads(inop), Exp=['ADS is
                         inoperable']; css(inop,Exp);
                         lpci(inop,Exp).
```

The Prolog version has two rules. This is because it was desired that the explanation ("Exp") be specific enough to denote which individual equipment being inoperable has resulted in the LCO. M.1 would provide the reason "because 3.5.1-2" if this rule fired; the reason given would be the same irrespective of what combination of system status resulted in the rule being true. In the Prolog version, the reason given would include which low pressure system(s) were not operable by virtue of the "append" operation, which is used to build up a detailed explanation. ("Append" is not a built-in function, but was programmed into the system using one rule and one statement. Two Prolog texts are listed in the References for readers interested in acquiring a further understanding of Prolog.)

The second general LCO handled by the Technical Specification Advisor is 3.0.5, which states:

> When a system, subsystem, train, component, or device is determined to be inoperable solely because its emergency power source is inoperable, or solely because its normal power source is inoperable, it may be considered OPERABLE for the purpose of satisfying the requirements of its applicable Limiting Condition for Operation, provided: (1) its corresponding normal or emergency power source is OPERABLE; and (2) all of its redundant system(s), subsystem(s), train(s), component(s), and device(s) are OPERABLE, or likewise satisfy the requirements of this specification. Unless both conditions (1) and (2) are satisfied, the unit shall be placed in at least HOT SHUTDOWN within 6 hours, and in at least COLD SHUTDOWN within the following 30 hours. This specification is not applicable in Conditions 4 or 5.

LCO 3.0.5 and the definition of OPERABLE (presented earlier) result in some of the most convoluted interrelationships within the Tech Specs. In the definition of OPERABLE, not only the given system but also its attendant electrical power supplies must be functional. The definition requires both the normal and the emergency diesel to be OPERABLE, even though the device will work with either one. Specification 3.0.5 gives the licensee an out, however, provided one of the power sources is available and all redundant systems (the other Train) are operable.

Without going into the basis for this specification, suffice it to say that it makes for some interesting rules. The appropriate rules from the M.1 version of the Technical Specification Advisor Pilot Project have been excerpted and appear below:

```
/*  The following rules arise from Tech Spec 3.0.5: */

     '3.0.5-1a':if edga = inop and ebusa = inop
                then power_a = none.

     '3.0.5-1b':if edgb = inop and ebusb = inop
                then power_b = none.

     '3.0.5-2a':if (edga = inop and ebusa = op) or
                 (edga = op and ebusa = inop) then power_a = half.

     '3.0.5-2b':if (edgb = inop and ebusb = op) or
                 (edgb = op and ebusb = inop) then power_b = half.
```

```
'3.0.5-3a':if edga = op and ebusa = op then power_a = full.

'3.0.5-3b':if edgb = op and ebusb = op then power_b = full.

'3.0.5-4a':if power_a = half and trainb = inop then lco =
          '3.0.5: Place unit in hot Shutdown within 6 hours
          and cold shutdown within the following 30
          hours.'.

'3.0.5-4b':if power_b = half and traina = inop then lco =
          '3.0.5: Place unit in hot Shutdown within,n 6
          hours and cold shutdown within the following 30
          hours.'.

/*  NOTE:  Train A/Train B definitions (part of 3.0.5)
           must be modified as more rules are placed
           into the knowledge base. */

'3.0.5-5a':if power_a = none or lpcia = inop or cssa = inop
          or slca = inop then traina = inop.

'3.0.5-5b':if power_b = none or lpcib = inop or cssb = inop
          or slcb = inop then trainb = inop.
```

First, the concept of train operability had to be addressed. This is accomplished via rules 3.0.5-5a and 3.0.5-5b in the knowledge base. Also, the exact condition of the power supplies to each train, both normal and emergency, had to be determined. This is accomplished via rules 3.0.5-1a through 3.0.5-3a and the corresponding "b" rules.

These latter rules, which result in the value of "power_a" and "power_b" (none, half, or full), are used not only in determining the 3.0.5 applicability but also in 3.8.1.1 and for each individual electrical load. This is because if there is no power to a train of components, all components in that train are obviously inoperable. Also, 3.8.1.1 describes what to do under various permutations of electrical power degradation.

The Standby Liquid Control System rules are straightforward, first because the Technical Specifications describe what to do not only for one pump inoperable but also if both pumps are inoperable and secondly because no other systems are referenced by the specification. It still takes six rules to adequately describe this specification. This is because the values of operability/inoperability which are input for the SLC pumps must not only be used in the 3.1.5 rules but also in the determination of whether or not Train A or Train B is operable. This same complexity arises for Core Spray and Low Pressure Coolant Injection.

Another form of interplay is demonstrated in the High Pressure Coolant Injection specification. HPCI can be inoperable for up to 14 days, as long as the Automatic Depressurization, the Core Spray, and the Low Pressure Coolant Injection systems are all operable. If such is not the case, the LCO does not say what to do; therefore, as stated above, specification 3.0.3 is invoked. Rules in the knowledge base accomplish this.

Without going into detail concerning every rule in the knowledge base, the above discussion at least provides the flavor of the process used to handle not only the straightforward LCO's and their action statements but also the interrelationships among them. It also bears mentioning that the goal "lco" of the knowledge base is multi-valued; it is important that all instantiations of this goal be conveyed to the user.

User Interface

The initial operator interface for the system was designed as a trade-off between ease of programming and ease of use by the operator. This was felt to be appropriate in the case of the pilot project, since satisfaction of the goals was not dependent upon development of a full-featured, user-friendly interface. The pilot program demonstrates the ability to interface between a database file and M.1 or Prolog. Future enhancements to the human factors of this part of the system would involve no new computer technology.

The system consisted of three modules: Input, the expert system, and output. A DOS batch file was utilized to sequence the advisor through these modules. Each module is described in more detail below.

Compiled BASIC was utilized to write a simple input system. The current status of plant systems addressed within the Technical Specification Advisor are contained in a BASIC random-access file, along with the allowed values of status (ie, "operable," "inoperable"). The operator is presented with several screens of plant systems and current status; he may toggle among the allowed values until the displayed parameters match the desired conditions. When the operator is through, the program updates the plant status database and writes the applicable facts to a text file in a format suitable for input into the expert system.

In the M.1 version of the knowledge base, the M.1 meta-fact "configuration(startup)=go" causes M.1 to immediately begin running the expert system when invoked. Likewise, the meta-fact "initialdata" is used in conjunction with the meta-proposition "do loadcache" to load in the values of the applicable plant equipment from the data file generated by the input module. At the end of M.1's inference, when the values for "lco" have been determined, the "do savecache" meta-proposition saves the results of the process, the cache memory, to an output file. This provides for a convenient method to control the output format.

The Prolog version is similar. ED Prolog allows an optional command line file name which indicates where the input stream is to come from. This feature is utilized to automatically consult the portion of the knowledge base which describes the Technical Specifications. The current plant status (the present "world view") for the consultation is loaded next from the text file generated by the input module. The goal "lco" is sought, with output to a file.

The BASIC output program sorts through the appropriate output file generated as a result of the expert system consultation. The program formats the LCO's generated and presents the result to the screen or printer. In M.1, the LCO rules were written in such a way that the label of each one contains the reference back to the section of

Technical Specifications from which it came. In Prolog, the rationale or explanation for concluding each LCO was implemented directly. This justification is provided to the Shift Foreman in addition to a listing of applicable LCO's.

Expert System Validation

As with any computer system, an expert system must be validated to ensure that the actual system design matches what is required and desired. In the case of the Technical Specification Advisor, which is a limited-scope pilot program, it would at first appear that an easy way to accomplish validation would be to exhaustively test all permutations of input data. However, even with such a small system, "combinatorial explosion" renders this infeasible.

There are seventeen systems or pieces of equipment at the input of this knowledge base. Even though each of these has only two permissible values, and ignoring the intermediate determinations made within the knowledge base, there are still 2^{17} combinations of operable/inoperable equipment. Therefore, a scheme is required to limit the number of test cases which will also provide knowledge base validation.

There turns out, however, to be a great deal of symmetry within the knowledge base itself. First, the existence of redundant trains of equipment leads to the first thumb rule for reduction of testing: if the rules in the knowledge base are symmetrically written, then a data test which tests one set of logical interactions effectively tests the symmetrical rules.

A second area which helps to reduce test cases is the independence of some of the Technical Specifications, coupled with the multi-valued property of "lco" in the knowledge base. Since the expert system was designed to search for all values of "lco," the test for several independent pieces of equipment may be incorporated into one test for all. Both of these test-limiting characteristics were utilized to narrow down the test domain.

A "Test Case Selection Matrix" was utilized to delineate the values corresponding to operability or inoperability for each system or component. Each case was designed to test a rule or series of rules. Nineteen cases comprising the matrix were run through the Technical Specification Advisor. The results were then checked meticulously by the Tech Spec human expert.

Extensive debugging was required as a result of the initial test case runs. It was discovered that, although it appeared relatively easy to express the 3.0.5 rules in "if-then" format, such was not the case. In a subsequent testing phase, however, only minor debugging was required. The results of the test cases on the current version of the knowledge base agreed with the human expert's results from the same tests. It was concluded that the knowledge base correctly modeled the Technical Specifications within the scope of the project.

A LOOK AT THE FUTURE

A number of expansion opportunities and potential enhancements to this project suggest themselves. The most obvious one is to expand the Advisor to a full scope system covering the complete Technical Specifications and all modes of operation. It is also considered essential to system utility to provide a "what if" mode, perhaps employing a separate database of plant status. A better, more sophisticated input/output system should be developed.

This expert system was designed to emulate the Shift Foreman. There is no reason to limit the scope in this fashion, however. One could develop a component level database system and a separate expert system to diagnose whether a system is operable or inoperable. This could store the knowledge of past engineering evaluations regarding system operability as part of the knowledge base. This system would serve as a front end to the Tech Spec Advisor described in this report.

Another potential for enhancement could be the addition of date/time stamping for LCO's generated by the Tech Spec Advisor. This could form the input into an automated LCO tracking system. A feature which could potentially be added is the ability to learn/update the knowledge base as Technical Specification interpretations are made.

Many of the potential enhancements are not expert systems per se, but rather more conventional programming applications. Nevertheless, they could result in a hybrid system capable of determining compliance actions and options available to limit entry into action statements.

CONCLUSION

Because of the complicated wording and interrelationships among various Technical Specification statements, compliance with the Tech Spec requirements has not always been perfect. In view of the potential safety and regulatory impact associated with non-compliance, a system which aids the Shift Foreman in Tech Spec compliance would be beneficial. The Technical Specification Advisor Pilot Project was undertaken to explore meeting this need utilizing expert system methodology.

The project scope was small enough to be performed with limited resources, but large enough to determine the viability of the approach. The goals of the project included demonstration of feasibility and determination of operator acceptance to the idea. Another goal was that the project provide the basis for a presentation to management with a view toward developing a full scale system.

Knowledge acquisition was facilitated by the author's knowledge of Standard Technical Specifications and the availability of an expert at the Brunswick site. Software was selected that would run on a personal computer; both M.1 and Prolog were utilized in the pilot project. The system was developed in an iterative fashion, using rapid prototyping, testing, evaluating, as three steps in a cyclic process to develop the knowledge base rules and facts.

The Technical Specification Pilot Project demonstrated the feasibility of a personal computer based Technical Specification Advisor for a small subset of Technical Specifications for one mode of operation.

For both M.1 and Prolog, this project demonstrated the ease with which a commercially available, personal computer expert system software may be utilized to build an expert system of value in assuring Technical Specification compliance. Considering that the avoidance of unnecessary fines or a reactor shut down can equate to hundreds of thousands of dollars, a closer look into developing a full scope Technical Specification Advisor is warranted.

REFERENCES

1. Bratko, Ivan. Prolog Programming for Artificial Intelligence. Reading, Massachusetts: Addison-Wesley Publishing Company, 1986.

2. Clocksin, W. F. and Mellish, C. S. Programming in Prolog. Berlin: Springer-Verlag, 1984.

3. Hayes-Roth, Fredrick; Waterman, Donald A.; and Lenat, Douglas B., ed. Building Expert Systems. Reading, Massachusetts: Addison-Wesley Publishing Company, 1983.

Computer Aided Modeling and Expert Systems Add a Needed Dimension to Water Management in Power Plant Operations

PATRICK H. GILL, JR.
Calgon Corporation
Water Management Division
P.O. Box 1346
Pittsburgh, Pennsylvania 15230, USA

ABSTRACT

Computer modeling and expert systems applications in water management are being used to develop appropriate treatment recommendations and monitoring/control functions. Chemical treatment program development in recirculating cooling water (CALGUARD), once-through scale inhibition (THRUGUARD), once through corrosion control (OSCAR), as well as internal boiler water chemistry control (POWER-CHEM), and on-line real time system monitoring (HELMSMAN) play an increasingly important role in power plant water management practices.

INTRODUCTION

Reliability and availability of power generating capacity continue to be primary goals in the power industry. Moves toward increased competition among power producers have already begun to curtail the resources available to plan, implement, and monitor water management practices at site locations.

Calgon Corporation has developed several types of computer aided modeling and expert systems that offer much improved utilization of available resources. These tools allow alternative water management approaches to be evaluated for effectiveness and optimization prior to implementation. Operating conditions and treatment results are being managed using on-line, real-time expert systems in cooling water applications. Boiler operations can also be effectively managed via an expert system developed specifically for control of water chemistry parameters.

Deriving a course of action from a wide range of alternate chemistries, application conditions, and desired results is no small task. When you consider the different experience levels among those involved with water management decisions as well as interpretation of "rules of thumb", it is no wonder that clear direction in scale and corrosion protection programs is an elusive commodity.

Calgon Corporation recognized the need for a systematic approach to alternative water management practices for use in designing such systems. "Rules of thumb" needed to be replaced with supported fact. The guiding principle behind Calgon's original program was the need to simulate operating conditions in order to predict the extent of water system problems and then to

predict the response of various treatment approaches. The result of Calgon's quest was the CALGUARD system, a mathematical modeling program used to make accurate predictions for recirculating cooling water scale, corrosion and deposition control. THRUGUARD is a similarly designed modeling system used to predict scaling potential in once-through cooling lake applications and the performance of various treatment alternative chemistries. Chemical treatment levels are recommended to inhibit scale in surface condensers in this manner.

The OSCAR system makes appropriate treatment recommendations for threshold corrosion control of once-through systems such as utility service water systems.

These predictive water management tools have eliminated trial and error and have greatly improved the reliability of treatment recommendations. Proper response to changing water qualities, as well as treatment optimization, can likewise be predicted.

Proper control of water management programs in cooling and boiler operation is essential to their success; expert systems have been developed to assist operators in making correct and timely control decisions. One example is an expert system utilizing on-line real time data in cooling water systems; developed jointly by Calgon Corporation and Texas Instruments, Inc. It continuously monitors the cooling water system for problems, ensuring that corrective action can be taken before a significant performance failure jeopardizes unit reliability. This system contains a great deal of knowledge represented in over 1,000 rules.

POWER-CHEM, a system developed by Mr. Leyon O. Brestel and Mr. Lon Brouse, is used to evaluate internal boiler water treatment of utility operations. Computer generated chemical treatment suggestions are made based on unit operating data. Graphics are continuously updated and are used to correlate system conditions for problem analysis. This system provides an effective tool for managing chemistry programs.

These expert systems have proven to be practical and very effective in the design and control of water management programs. The task of incorporating new rules and expanding the application of these systems continue. The use of expert systems has greatly improved the quality and reliability of the systems being served.

<u>CALGUARD</u> (Recirculating Cooling Water Modeling)

The concept of modeling as it relates to CALGUARD describes the process of capturing essential relationships of an on-line cooling water system in mathematical terms. These terms take the form of equations used to simulate the operation and interrelationships encountered in an operating system.

As you might expect, the number of independent variables in a cooling water system is very large. It is important to define critical variables, as those which influence the response of dependent variables to the greatest degree.

In the area of corrosion control, some of the critical variables are pH, temperature, and water chemistry. In the area of scale inhibition, critical considerations include pH, temperature, and the degree of saturation of scaling salts. For deposit control, variables such as pH, water velocity, and heat flux represent major parameters. In all these areas, additional variables as well as the amount of treatment chemical and the nature of the chemical are critical to the response observed in the dependent variable.

After the experimental series have been designed and tests performed in the development of CALGUARD, the data is mathematically analyzed. This analysis takes the form of putting levels of the independent variables, and values for the measured responses, through a mathematical algorithm which interprets interactions of independent variables as they affect dependent variables. The result is a mathematical equation which is used to represent the data.

This is a multiple-regression equation that actually describes what is called a response surface. One can visualize that if two variables were measuring a response as a function of two independent variables, all of the answers would be on a two-dimensional surface. If the response were linear in all directions, this response surface would be a plane. If the responses were non-linear, their surface would have hills and valleys; if you connected all the points with similar responses, the resultant graph would look like a topographical map.

Such a map is called a contour plot and is useful in determining such things as conditions for optimum corrosion control, or the best blend of two chemical building blocks. By fixing all but one of the independent variables, we can get a two-dimensional cross-section of this response surface and plot things such as corrosion rate versus inhibitor level, or percent deposit inhibition versus treatment level. If desired, we could also plot dependent variables versus a range of any independent variable investigated. The net result is that the value of the dependent variable anywhere on the response surface can be examined with confidence.

If you were dealing only with the case of two variables having linear responses the resultant equations would be fairly simple, and a hand calculator plus a little patience is all that would be required to solve them. However, in the Calguard system we are dealing with equations of high complexity, some having as many as 30 terms, with some terms being the result of other equations, a powerful computer is required to obtain solutions.

In CALGUARD these equations are handled routinely, along with problems in differential and integral calculus and the solving of many simultaneous equations. When this complexity is combined with the ability to examine a variety of treatments, the result is a very powerful tool for treatment program development, and system analysis that can only be accomplished using high speed computer capability.

There are several separate models in CALGUARD which are tied together during a program run. One is the calculation of material balance relationships of cooling systems, used to determine physical parameters of a cooling water system, such as make-up requirements and system half life.

CALGUARD goes beyond simple models to more arcane considerations. One unique model employed is termed the "water quality model," used to take any given makeup or recirculating water and, by considering levels of its chemical constituents, determine corrosion aggressiveness of the water on various metals. Another sophisticated model employed by CALGUARD is called the "solubility model". The solubility model concerns itself with soluble salts in cooling water. This model examines system operating conditions and thermodynamic activity coefficients for each dissolved specie in the water for ion pairs. In total, 18 ion pairs and 16 species are considered.

Unique to the field of water treatment each of the equations can consider up to thirty terms. Approximately twenty of these equations are solved simultaneously in order to predict accurately the saturation levels and the scale inhibition effectiveness of chemical treatments.

Within the CALGUARD program over ten models are used to simulate and predict water chemistry performance unique to any given cooling system. Accuracy of these models has been demonstrated in over ten years of use and application experiences.

Unlike computer "data banks" which are simply historical collections of disjointed experiences, CALGUARD is based on fundamental technology generated through designed experimentation. It contains performance data on all the major chemical building blocks used in the treatment of cooling waters, covering wide ranges of operating and water quality variables. With this capability, CALGUARD has the ability to accurately predict cooling water system performance, anticipate problems before they occur, present viable treatment options, and determine the optimum treatment program needed to meet maintenance, production, and profitability goals.

<u>THRUGUARD</u> (Once-Through Scale Inhibition Model)

The potential for calcium salt scale and deposition is as much a concern to power plants using once-through cooling water as it is to those using recirculating cooling water systems. Once-through cooling represents a unique set of conditions very different from those used in the determination of treatment considerations for recirculating systems.

CALGUARD uses thermodynamic constants to evaluate treatment options for recirculating cooling water systems. Recirculating cooling systems can be controlled to maintain desired cycles of concentration; system water residence time can be measured in hours and days during which time equilibrium conditions are established. Chemical treatment objectives aimed at 100% scale inhibition in recirculating systems can easily be achieved on an economical basis due to controlled losses in these systems. In contrast, consider scale inhibition of once-through cooling water systems with condenser residence time of only 7-10 seconds. A continuous supply of inhibitor is needed since there is no recycled chemical residual as in recirculating systems; thus, cost effective treatment is a challenge.

The THRUGUARD computer modeling program was developed specifically for once-through cooling scale inhibition. THRUGUARD uses all of the same rigorous solubility calculations as in CALGUARD; in addition, it looks at residence time as a function of nucleation and crystal growth. Thus, the degree of scaling potential and the inhibitor level necessary to inhibit scale are determined from kinetic considerations as well as from basic thermodynamic solubilities.

In most analytical procedures, the concentration of the desired species is determined and expressed in a weight ratio (e.g. parts per million parts). This is a measurement of all of the selected ions present. However, those ions that, at any moment, are participating in ion pairing are not available for other reactions. The more precise term, "activity", is used to describe only those ions that are freely circulating in solutions and available for reaction. This is why one frequently sees the situation where, by analysis, the water is apparently supersaturated with calcium carbonate, but, in reality no precipitation occurs. In other words, while the concentration of calcium and carbonate ions are in excess of the solubility product for these ions, their activity in solution is something less and the solution is stable.

The THRUGUARD program takes a complete water analysis and performs a series of complex calculations to determine the ion activity product (IAP), the solubility product constant (Ksp), and several other parameters for each of the scale forming salts of interest. Saturation level and ppm over equilibrium calculations are based on calcium and carbonate ions in water (in the case of calcium carbonate scale). This is a smaller number than the

concentration as determined by analysis. The saturation level is the IAP divided by the Ksp and reflects the number of times that the active species exceed the solubility product.

Momentary excess is calculated including all species in the analysis. This indicates the potential scale forming molecules present in water in either free or ion paired form. As a rough approximation, the difference between "ppm over equilibrium", and "Momentary Excess" can be considered the amount of ions participating in the ion pairing.

The recommended variation in chemical inhibitor dosage is calculated by THRUGUARD based on the known performance of each product as a function of pH, temperature, retention time in the system, and the saturation level in the water. In this manner, scale inhibitors are accurately applied to once-through systems in a cost effective manner.

Seasonal conditions and differences in water quality, pH, and temperature are routinely monitored. Scale inhibitor dosages are continually optimized based on system requirements as predicted by the THRUGUARD modeling program. The validity of this program has been proven through nearly ten years of application experience and well over 100 system years of use in the power industry.

<u>OSCAR</u> (Once-Through Corrosion Control Expert System)

Corrosion control in once through systems is frequently considered an art, where decisions are made based solely on guesswork and personal experience rather than on supported fact. In an effort to eliminate trial and error involved with the application of corrosion inhibitors in once-through applications, an expert system for treatment recommendations is employed.

OSCAR utilizes an IBM expert system shell program that resides on a main frame computer, accessed via modem and a telecommunication network.

The OSCAR program considers water quality, system parameters, and treatment objectives using over 500 rules in order to develop an appropriate treatment recommendation. The rule base is an accumulation of knowledge on once through threshold corrosion technology developed through fifty years of Calgon application expertise in this area.

The Expert System is a multi-level, multi-faceted decision analysis tool using information specific to the system in question. The program provides a systematic approach to the selection of treatment chemistry based on the performance of various building blocks under specified conditions. Treatment objectives of once through systems are taken into consideration in the analysis. Those objectives are summarized as follows:

1. Corrosion control
2. Corrosion product deposition control
3. Iron/manganese fouling control
4. Silt and microbiological deposit control

The primary advantages OSCAR provides is uniform application decisions regardless of operator experience level. The program assures that ineffective chemistry will not be used in any situation.

<u>HELMSMAN</u> (Cooling Water Expert System)

The water treatment industry has made significant strides in the area of automation and control. Automated blowdown control based on conductivity, flow proportioned chemical feed and pH controlled acid feed have been widely accepted. Still, analytical data collection is primarily a manual effort.

Test data is collected, logged and filed away. The data often stays as data, with all-too-little transformation to information upon which better control decisions can be made.

Constructing a control history from log sheets is difficult at best and generally a last resort. There is often no easy means to access data for troubleshooting or fine tuning. Alarm conditions so subtle that an experienced eye for chemistry is required often go unacknowledged with no corrective action. Reactionary control is the norm. Over control is as prevalent as under control. All too often the sure sign of serious control deviation is equipment failure or forced outage. Problem solving and diagnosis will depend upon the experience and knowledge at hand. Misdiagnosis and the wrong corrective actions may be taken, exacerbating the problem or delaying its resolution. In general, industry is being deluged with data about processes but there is difficulty with its assimilation, correlation and integration into the decision making process. Many believe that the industry has exhausted most of the engineering and mechanical improvements that can be made in the existing process. Further incremental improvements often require deployment of human resources more productively. An opportunity exists in the water treatment industry to address the fact that human expertise in solving day-to-day water treatment problems is an increasingly rare and valuable resource, and that a widening knowledge gap exists on how to successfully treat cooling water.

In early 1986, Calgon Corporation and Texas Instruments Incorporated entered into a project to capture the knowledge of cooling water experts and to place this knowledge into an on-line, real-time expert system that could continuously monitor water treatment effectiveness and diagnose the system if problems appeared. The system's development was completed at the onset of 1988.

Artificial intelligence (AI) is that part of computer science concerned with designing intelligent computer systems that exhibit the characteristics associated with intelligence in human behavior. In essence, AI is the study of how to make computers do things at which people are better. Expert, or knowledge-based systems is the branch of AI that deals with ways of representing knowledge in a particular domain (area of expertise), consisting of facts about the domain and symbols and "rules-of-thumb" (Heuristics), rather than numbers, for applying those facts.

The Expert System continuously monitors the cooling water system for problems, ensuring that corrective action can be taken before a significant performance failure jeopardizes unit reliability. The expert system contains a great deal of knowledge represented in over 1000 rules; however, it operates on a PC class device. Rules are processed with a control algorithm called the inference engine. The user (operator, supervisor, middle manager) views diagnostics from a user interface and can implement corrective actions. The diagnostic system evaluates present and past water chemistry and corrosion data in relation to system characteristics and chemical treatment strategy, identifies out-of-limits conditions, and lists in order of decreasing certainty the probable causes of the out-of-limits conditions. Assignment of certainty factors depends on the various combinations of data that point to the various conclusions. Since the system runs autonomously, it was designed to produce the best analysis possible with the data available. A forward chaining technique was chosen to examine this varied assortment of data and to draw conclusions in the form of diagnoses. Diagnoses are accompanied by action statements to provide insight into the actions that are needed to return out-of-limit conditions to normal or actions that are needed to more fully understand the problem.

The system accepts real-time data as required from available on-line sensors. In the ideal application, all water data input to the expert system would come from sensors, providing maximum awareness of system status. Advances in instrumentation for cooling water applications make many of the data inputs to the expert systems available via on-line sensors/analyzers. To accommodate sensory technology that may not exist or be in place, analytical data obtained through other off-line sources can be entered manually at the user interface. Historical data is maintained to determine trends. All data is time-tagged to determine whether it remains adequately current for its intended purposes.

The system is flexible and has a broad range of knowledge, applicable across a wide range of recirculating cooling water scenarios. It can be customized for different installations and treatment strategies, and can grow as new treatment chemicals and methodologies become available. As the opportunity arises for the rule base housed in the system to expand, new expertise is added to the program. The system is robust and easy to use in order to accommodate a user community consisting of a wide range of skill levels.

EXPERT SYSTEM HARDWARE - To facilitate the goal of on-line, real-time operation, the approach taken was to have the expert system run on a dedicated hardware device. The module chosen is an MS DOS-compatible 80186 processor with 1 Megabyte of RAM (random access memory), I/O ports, ROM (read only memory) disks, and a built-in MS DOS operating system. The module has access to all of the facilities of a programmable logic controller to which it is connected when used in an automation and control mode. The module is connected via a serial link to an IBM AT class computer that serves as the user interface and is typically dedicated to other data management activities.

SOFTWARE - Software development was initially attempted using a commercial expert system shell. But because of the large size of the knowledge base, the need for numerous custom I/O formats with help facilities, the heavy emphasis on calculation and record keeping, and the memory size requirements, a custom design was determined to be more appropriate. An extendible (yet non-LISP) language that would operate efficiently on the module was chosen to meet the goals for size and comprehension by experts as they expand the system. Using an extendible language permits the rules to be written in a language customized to the domain and easily understood when read by the expert. The system uses non-preemptive multitasking to perform new analyses while the user is either inspecting available reports or has disconnected the user interface from the expert system's dedicated module.

So that people of varying levels of computer and water treatment skills can easily use the system, a customized, windowed environment was created to provide reports and access to voluminous numerical data for display or editing. Help windows are available to assist in filling in various data fields and for interpreting data and analyses that are displayed.

Key hardware accompanying the expert system's dedicated hardware module includes a programmable logic controller and associated I/O modules that connect to the sensors and actuators of the cooling water system. The typical installation will have the programmable logic controller and the expert system's dedicated hardware module both plugged into the RS232C links of the user interface (IBM AT class computer). The programmable logic controller handles such tasks as chemical feed, blowdown control, and pH control. It gathers data from the on-line sensors so that it is continuously available to the expert system's dedicated hardware module, which is plugged in the backplate with the other modules. Other process control software resides in the user interface providing large amounts of non-diagnostic historical data recording, trending and analysis.

The dynamics of large cooling systems can be quite slow. This means that performing an analysis of the system once per hour offers a good balance between providing adequate current system diagnostics and restricting the amount of on-line historical data and diagnoses to a manageable size. Once per hour may not sound like "real-time", but the expert system completes a diagnosis in about three seconds and would consequently be acceptable for applications with much shorter time constraints.

This system provides a reliable decision making tool given its diagnostic approach. As a decision aid the probability of fixing it right the first time is enhanced. Emerging problems can be identified as they occur, not after damage becomes a visible symptom. The expert system can play a role in maximizing the effectiveness of available resources, including operators, supervisors, and middle managers. Calgon HELMSMAN facilitates the data collection, interpretation and decision making tasks, allowing precious time to be more focused on corrective investigations and actions. Expert systems used in water management applications can serve to stimulate plant personnel to initiate routine and fundamental corrective changes directly, and handle more responsibility in a quality fashion.

POWER-CHEM (Boiler Water Chemistry Expert System)

Boiler chemistry is inherently unstable. Items such as inleakage of air and cooling water into the condenser, unit load patterns, makeup water system performance, previous chemical control decisions, chemical feed system operation and reliability, and analytical bias combine to produce the "current state of chemistry" that are described with analytical data obtained from grab sampling and on line analyses systems.

By introducing a microcomputer-based system to model chemistry trends, the chemist can simplify his task considerably. The desk-top computer system provides assistance in controlling boiler pH and phosphate levels within specified limits for sodium/phosphate congruency. The result is a computer-recommended chemical dosage and the boiler-blowdown ratio necessary to balance the specified residual.

Consistency and stability are designed into the program logic by managing key parameters as constants. These include chemical additions, chemical tank drawdown rates, and blowdown valve settings. The predictive-control concept, coupled with computer evaluation of chemistry trends, is used by the plant chemist to manage boiler-chemistry data and to help identify and control root problems. The goal is to avoid reactive chemistry control, which can result from making control decisions for an out-of-balance system without adequate evaluation of the situation. This leads to follow up control, which is often based on response to the last fix.

Predicated on the ability to mathematically model the system, predictive control avoids those pitfalls with one-step corrective measures. Moreover, computer comparisons of recommended chemical feeds to previous chemical additions, and their relationship to the sodium/phosphate ratio of boiler water, expose rough data and external influences. An expert computer based system will expose upsetting influences and recommend appropriate chemical corrections.

Current data is entered daily into the computer. Data entries include: date and time, drum pressure, unit load, pH, alkalinity, conductivity, phosphate, and silica; recorded data taken at the time of sampling is also included. This is followed by chemical addition and blowdown data: Chemical feed tank delivery of treatment chemicals, blowdown valve setting, and blowdown flow, if available.

Approximately 20 data points are involved; they are instantly available for viewing on monitor charts which can quickly expose analytical and operational anomalies. Trend plots are used to examine previous chemical additions to the system. Contaminating influences and consistency of chemical additions are discernible for comparison to plant operating trends.

Boiler blowdown can be evaluated against computer-generated silica control curves as well as water-loss determinations from mass balances. From these data, the adequacy of the existing blowdown rate is considered for new chemical feed specifications.

Defined responsibility, management support, and accountability are central to the continued quality of a chemistry program. Performance goals should be defined, and progress evaluated against those goals. Computer generated performance charts, showing the percentage of observed values falling within control limits, help identify the areas of greatest difficulty and provide a focus for technical support.

The POWER-CHEM software can help the control process, but even this expert system would result in reactive control if inaccurate, unverified data were included in the database. Logical decisions based on inaccurate input become illogical. Quality assurance of all laboratory and chemical-control procedures should be the operating foundation for every power plant chemistry program before expert systems for boiler chemistry can be of value.

Conclusion

The use of modeling programs for the design of water management practices along with expert systems used in monitoring and control of treatment application is proven and available technology.

The net result from our experience to date is that more informed decisions are being made in the development of water management recommendations. Application control is very much improved helping to yield consistent results for which chemical treatment programs are designed. Areas in need of improvement are exposed and root problems can be addressed.

These modeling and expert systems are expanded as new data becomes available . They represent tools that help meet water management needs of the power generation industry today with the capability of helping meet the growing challenges ahead.

REFERENCES

1. S. D. Strauss - Expert System Helps Fine Tune Boiler Water Chemistry, October, 1987 issue of Power Magazine.
2. David Roberts, Richard Herrod, Leon Tietz - "On-Line Real Time Expert System for Cooling Water," presented at the 49th Annual Meeting of the International Water Conference, Pittsburgh, PA - October 24-26, 1988.
3. "CALGUARD System Overview" - Calgon Product Literature - 11-109.
4. THRUGUARD Internal Calgon Communication by W. R. Hollingshad - 10/8/85.
5. OSCAR interview with program developer J. Harrison of Calgon 3/27/89.

On the Application of STARRS Methodology to Assess Tube Rupture Consequences in PWR Plants

A. T. WASSEL, S. M. GHIAASIAAN, and J. L. FARR, JR.
Science Application International Corporation
21151 Western Avenue, Torrence, California 90501, USA

S. P. KALRA and D. CAIN
Electric Power Research Institute
Nuclear Power Division
3412 Hillview Avenue, Palo Alto, California 94304, USA

A. SURI
Productivity Implementation Associates
23 Backus Avenue
P.O. Box 3613, Danbury, Connecticut 06813, USA

ABSTRACT

A mechanistic model and the associated computer code, which simulate the transport, removal and release of radioactive iodine during steam generator tube rupture (SGTR) events in a PWR plant are described in this paper. Steam generator tubes, which are prone to corrosion and mechanical damage can leak and/or rupture and, therefore, can result in the release of radioactive fission products to the atmosphere. A mechanistic computer code is therefore needed to quantify the magnitude of such release, and to calculate the associated activity dose.

The computer code, Secondary-side Transport And Retention of Radioactive Species (STARRS) simulates the transport and removal of radioactive vapor and aerosol species in the secondary-side of a U-tube steam generator. The code has been validated using the Model Boiler (MB-2) experiments and its application to calculate releases from typical plants as a function of operating thermal-hydraulic conditions is demonstrated.

This paper provides an overview of theoretical basis, the applications in calculating SGTR consequences and describes the user-friendly features of STARRS. This includes the menu driven input, on-line help, memory-resident graphics capability, and compatibility with personal and mainframe computers.

The interactive PC version of STARRS, which is enveloped under the general guidelines of the EPRIGEMS expert system, is described herein. This version of STARRS provides the user with appropriate guidelines on how to use the code, and how to analyze the results. Screen-sensitive help is also provided to assist the user in interpreting the input/output files, displaying the key results ensuring an error-free input data and in support of diagnostic decisions.

Both thermal-hydraulic aspects and radioactive retention aspects have been assessed using experiemental data. The code predictions are in good agreement with the data. In addition, the STARRS application has been extended to PWR plant simulations and a sensitivity analysis has been completed.

INTRODUCTION

Corrosion and mechanically induced damage can result in the rupture of the PWR steam generator tubes. Such a rupture will result in the leakage of the radioactive primary-side fluid to the secondary-side of a steam generator. A fraction of the leaked primary-side radioactive species will be retained in the secondary-side and the rest will be released in the form of vapor and entrained aerosols.

The occurrence of Steam Generator Tube Rupture (SGTR) events has long been recognized by the Nuclear Regulatory Commission (NRC). Operating limits on the radioactivity levels in the primary-side coolant have been imposed such that, in the event that a full tube rupture occurs, the activity release will not exceed the allowable limits of 300 rem to the thyroid in 2 hours (for iodine) [1].

A mechanistic model is therefore needed to quantify the amount and composition of the release during the SGTR events. Inadequate understanding of the physical phenomena and transport processes involved may result in the use of conservative limits, with unneeded restriction on reactor operating conditions. On the other hand, higher releases could impose stricter limits on the maximum allowable concentrations of radioactive species in the primary-side fluid.

The STARRS computer code was developed to quantify the amount and make-up of the radioactive species during an SGTR event. The code, which simulates both covered and uncovered tube ruptures, can be applied to both design-basis and beyond design-basis events. An overview of the theoretical basis of the code is provided in this paper.

An interactive Personal Computer (PC) version of STARRS has been developed to provide the user with appropriate guidelines on how to apply the code, as well as, on how to analyze and interpret the results. Through the use of this user-friendly STARRS code, an operator/analyst can analyze the radiological consequences for SGTR events. STARRS uses the EPRIGEMS artificial intelligence package [2], known as SMART. Additional interactive features include on-line help, memory-resident graphics capability, display of key results, display of SG schematics, including transient operating conditions, and appropriate data cross-checks for ensuring error-free input data.

This paper is structured as follows: first an overview of the modeling approach is provided, followed by a description of EPRIGEMS-STARRS software. Hardware requirements are then described, followed by a description of code applications.

MODELING APPROACH

A schematic of the secondary-side of a U-tube steam generator during an SGTR event is provided in Figure [1]. The key components and associated transport processes are depicted in the figure for a covered tube rupture. The STARRS computer code includes models for the key heat and mass transfer processes. The modeling approach is based on dividing the secondary-side into several zones in series; the exit conditions of a given zone represent the inlet conditions of the succeeding zone. Conceptually, the secondary-side is divided into the following: (1) break flow zone, (2) bubble rise zone, (3) swell level zone, (4) steam volume zone and (5) separator and dryer zone. In each zone, the appropriate governing differential equations are solved, rendering the temperature, mass and mass species of vapor and aerosols. The code can be exercised in the steady-state or transient mode. Furthermore, it simulates gaseous and aerosol trace species, treats polydisperse aerosols, simulates covered and uncovered ruptures, and treats a dry, partially full or full secondary-sides.

The details of the modeling approach can be found in reference [3]. The following provides an overview of the key phenomena simulated in STARRS.

FLASHING, ATOMIZATION AND BUBBLE FORMATION

A tube rupture can be of the guillotine-type or a fish-mouthed break. Upon rupture, the primary coolant liquid which is normally under 150 atm pressure flows into the secondary-side. The primary liquid partially flashes, creating bubbles, and partially mixes with the secondary-side liquid. A fraction of the primary

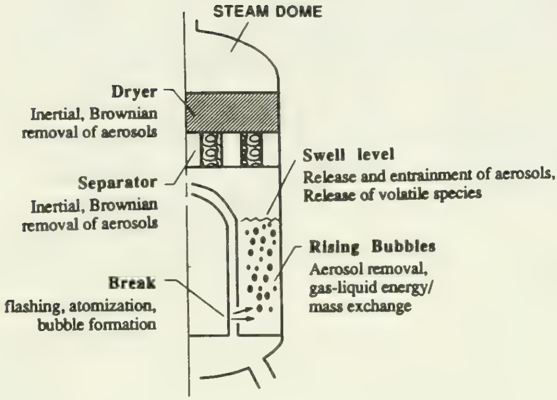


FIGURE (1) SCHEMATIC OF THE SECONDARY-SIDE MODELING ZONES

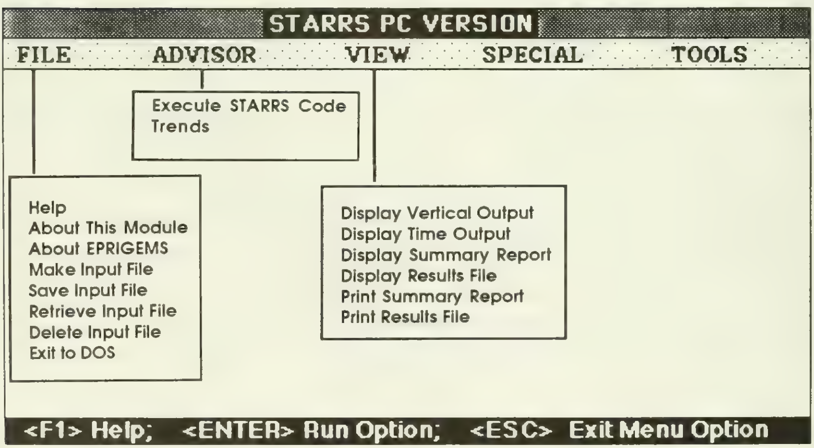


FIGURE 2 STARRS MENU-MAP

liquid forms liquid aerosols which are entrained by the formed bubbles. These bubbles, therefore, contain volatile species which are in the form of gas mixed with steam, or liquid species which are dissolved in the entrained liquid aerosols.

The flashing model assumes that the superheated vapor, originating from the primary-side fluid, loses a fraction of its superheating with respect to secondary side pressure, before stable bubbles are formed. The non-evaporating primary liquid is assumed to be atomized and entrained in the bubbles. The entrained aerosols are assumed to have an initial log-normal size distribution. The log-normal distribution of droplets resulting from the break-up of a jet is supported by the investigat on of reference [4]. The mean droplet diameter is calculated from a critical Weber number of 12.5, and a hydrodynamic break-up criterion is used in the analysis [5,6]. The flashing bubbles, upon formation, are assumed to be of uniform size, and have a maximum stable bubble size according to Lehrer [7].

If the break location is exposed during the transient (when the water level on the secondary-side drops below the break location) the primary coolant partially atomizes. The liquid aerosols which are too large to be entrained by the steam fall back into the liquid pool, and the smaller droplets are carried by the steam flow.

TRANSPORT AND REMOVAL DURING BUBBLE RISE

The bubble rise velocity is calculated using Harmathy's correlation [8]. The retarding effect of the surrounding pipes on the bubble rise velocity is included by using correlations suggested by the same author. As they rise, the bubbles exchange energy and mass with the surrounding water. The energy, mass and radionuclide species conservation equations are solved for the rising bubble swarm. These equations are cast in the form of ordinary differential equations representing the rate of variation of gas enthalpy, bubble total mass, and radionuclide species mass fraction. The resulting coupled, ordinary differential equations are then numerically integrated up to the swell level.

As bubbles rise in the liquid, the entrained aerosols undergo several removal mechanisms. Inertial deposition takes place because larger droplets cannot follow the gas stream lines. The gas in the rising bubble undergoes circulatory motions, which result in inertial deposition of the entrained aerosols. This mechanism is most effective for the removal of larger droplets. Sedimentation due to the

effect of gravity also takes place where gravity imposes a retarding force on the aerosols. As a result, some of the aerosols are deposited onto the liquid boundary of the bubble. Brownian diffusion, which is effective for removing smaller particles can take place, and results in particle diffusion towards the bubble surface. The convective flow of gas across the bubble liquid-gas interface gives rise to the convective deposition mechanism. In the case of a bubble rising in a liquid environment where evaporation takes place at the interface, the convective flow is away from the interface and towards the bubble center. In such a case, convection of gas has a retarding effect on the removal of aerosols. In case of condensation at the interface, however, the opposite takes place and the deposition of aerosols is markedly augmented by the convective flow. Thermophoretic deposition can also take place since aerosols move down the temperature gradient, and can deposit on the colder bubble gas-liquid interface.

TRANSPORT AT THE SWELL LEVEL

Two different streams reach the swell level -- the steam produced by boiling, and the steam carried by the rising bubbles which are formed at the break. The flashing bubbles contain aerosols (which have survived during bubble rise in the water) and vapor trace species. When these bubbles (whether produced by flashing or by secondary-side boiling) rupture at the surface, droplets are formed and carried by the steam flow into the separator. Surface entrainment is modeled, following Kataoka and Ishii [9]. The semi-empirical correlations of Kataoka and Ishii are used for predicting the rate of entrainment, size distribution of the entrained droplets, and the initial velocity of the droplets at the water surface.

REMOVAL IN THE STEAM VOLUME, SEPARATOR AND DRYER

When the water level in the secondary-side falls below the separator, the aerosols have to pass through a volume of steam before reaching the separator. In this region, the aerosols undergo removal by sedimentation and Brownian diffusion. Brownian deposition takes place on available solid surfaces. For example, when the tubes are partially covered, the aerosols can deposit on the tube bundle. To calculate removal by sedimentation, the aerosol equations of motion are solved using either the Stokes or high Reynolds number drag coefficients.

The dominant removal mechanism in the separator is inertial deposition, which is induced by the centrifugal force caused by the curvilinear motion of the particles. Brownian diffusion can also be significant because the steam flows through relatively narrow passages, and deposition on existing solid surfaces can

take place Mechanistic models are developed for the inertial deposition in the separator. The curvilinear equation of motion of an aerosol in the separator passages is solved, and the rate of deposition is calculated. These equations of motion are solved for each aerosol size and type. The Brownian deposition rate is calculated by using analogy between aerosol deposition and heat and mass transfer.

The dryer in a PWR plant has the configuration of chevron plates. The removal mechanisms of significance are, again, inertial and Brownian. Both mechanisms are modeled. For the inertial deposition there are two options. One option uses a mechanistic model which is based on a solution of equation of motion of aerosols in idealized chevron passages. The other option uses a semi-empirical correlation [10]. Brownian deposition is modelled using an analogy between aerosol transport and heat and mass transfer processes.

DOSE RELEASE

The amount of iodine released from a steam generator is dictated by the mass fraction of iodine in both the steam and liquid spaces on the secondary-side. The iodine release rate depends on the accummulation rate of iodine vapor in the steam space above the swell level, as well as, on the concentration of iodine which is dissolved in the water below the swell level. The STARRS computer code solves the transient conservation equations, which govern the mass fraction of the iodine species in the steam and liquid spaces.

The key parameters that are calculated by the code include: (1) the differential decontamination factor DDF (ratio between the mass flow rate in and the mass flow rate out), (2) the integral decontamination factor IDF (ratio between the cumulative mass in and the cummulative mass out), (3) the mass of vapor and liquid iodine leaving the steam generator, (4) the rate and total radioactivity release for I-131, and (5) the ground level dose for given atmospheric conditions and windspeed.

STARRS SOFTWARE & HARDWARE

The EPRIGEMS project is a new initiative at EPRI intended to streamline the process of capturing and conveying R & D results in directly usable form to our utility members. EPRIGEMS products harness new computer technologies, such as expert systems, to convert utility personal computers into personalized problem-solving tools for applying EPRI research results. EPRIGEMS is neither a computer code in the traditional sense, nor a text-based information medium. The strength of the EPRIGEMS concept is its emphasis on providing guidance to the utility user.

STARRS has been embedded in an expert system shell (SMART), not totally for the purpose of accommodating EPRIGEMS interface. The expert system provides a future pathway for enhancing STARRS beyond the capabilities of conventional software. Examples include: computer-assisted selection of input parameters, intelligent results analysis, reasoning about plant design and operational modifications options to mitigate SGTR events, etc.

STARRS PACKAGING SYSTEM

EPRIGEMS provides a user-friendly environment in which end-users of the STARRS code can create and edit input files, execute the code, view the various output files, and finally create graphical displays of the key output variables using a sophisticated scientific graphics package. Some of the salient features of this system include: customized design, screen-sensitive help, validation of input to ensure error-free data, and a knowledge database to aid the user in assessing and interpreting the results.

The EPRIGEMS interface has been custom designed to provide the STARRS code user with more power and flexibility through the use of pull-down menus, a well-thought out color schematic to draw user attention to options available at various stages of processing, and a full screen editing capability to enable the user to freely move around the input screen and to save time during data entry.

Running STARRS can be a complex knowledge-based task. The extensive use of on-line, screen-sensitive HELP makes it easy for the user to recall information in order to fill an input field or make a decision as to which variable to plot. Since HELP can be obtained almost anywhere with the push of a button, no lengthy manuals are required.

Input preparation and data entry are the most tedious and error-prone steps in using any computer code. Through the use of a set of expert system rules, validation checks are made on each of the input variables to ensure its reasonableness. Another set of rules provides built-in cross-checks to validate interrelated data.

A comprehensive knowledge database provides the user with an on-line capability to assess and interpret the results generated by the STARRS code. This package provides the user friendly features and necessary advice/help for easy application of such a complex methodology to evaluate SGTR consequences in a PWR plant.

MENU OF USER OPTIONS

Upon execution of the program, the user is presented with an introductory screen which consists of a menu bar at the top of the screen and a set of additional key operations at the bottom. (See Figure [2]). There are five user options available on the menu bar: FILE, ADVISOR, VIEW, SPECIAL, and TOOLS. The key operations that appear at the bottom of the screen are: <F1> HELP, <RETURN> RUN OPTION, and <ESC> EXIT OPTION. The user selects one of the five options available on the menu bar by pressing the <ENTER> key. The user is then presented with a pull-down menu of additional options. Each of these options is discussed below.

The FILE option contains introductory information and file manipulation routines. This option provides the following choices.

1. HELP - contains information on how to use EPRIGEMS key selection, and how to move around in the menus.
2. ABOUT THIS MODULE - description of the STARRS code and who to contact for assistance in using the code.
3. ABOUT EPRIGEMS - description of EPRIGEMS and its use for technology transfer.
4. MAKE INPUT FILE - allows the user to create or edit an input file. (See Figure [3]).
5. SAVE INPUT FILE - saves the created or edited input files for use by the STARRS code.
6. RETRIEVE INPUT FILE - retrieves an existing input file so that it can be edited.
7. DELETE INPUT FILE - deletes an input file that is no longer needed.
8. EXIT - exits the EPRIGEMS program and returns the user to DOS.

The ADVISOR option provides the user with a way of running the STARRS code and a method for analyzing the data trends. This option consists of the following choices.

1. EXECUTE STARRS CODE - once the input file is created this option executes the code.
2. TRENDS - provides the user access to the knowledge database to interpret trends in the results and provide information on how to make adjustments to various input parameters to obtain desired results.

The VIEW option, usually selected after code execution, provides the user access to the various output files created by STARRS. This option consists of the following choices:

1. DISPLAY VERTICAL OUTPUT - displays output parameters as a function of spatial coordinates along the flow path between the break and the steam generator exit. Used in conjunction with the plotting routine.
2. DISPLAY TIME OUTPUT - displays output parameters as a function of time. Used in conjunction with the plotting routine.
3. DISPLAY SUMMARY REPORT - displays the summary tables on the screen.
4. DISPLAY RESULTS FILE - displays the complete output file on the screen.
5. PRINT SUMMARY REPORT - prints the summary report on the printer.
6. PRINT RESULTS FILE - prints the complete output file on the printer.

At the present time the SPECIAL and TOOLS options contain no choices and are not selectable by the user.

STARRS HARDWARE REQUIREMENTS

In order to use the EPRIGEMS, STARRS system, the user needs an IBM PC; XT, AT or any personal computer that is 100% compatible with these IBM models. In addition, the computer must be equipped with a math coprocessor in order to run the STARRS code. It is possible to configure the code to run without the math coprocessor, but that option is not really feasible, since the code would run an inordinate amount of time to be of any practical use. The computer must have at least 512K of RAM and preferably 640K. The operating system must be PC DOS, version 3.0 or later. The PC must have at least two floppy disk drives or a hard disk. A color monitor is desirable, but optional. To obtain a hardcopy of the code output or the plots generated by the plotting program, a printer is needed. A printer is not absolutely necessary, since options exist to display any of the results to the screen. An option to run the code interactively on a VAX computer is desirable; such an option has been implemented for a version of the STARRS code with EPRIGEMS features.

STARRS APPLICATIONS AND RESULTS

The evaluation of SGTR consequences requires modeling of thermal-hydraulic behavior and activity transport behavior. There have been a large number of codes developed to simulate thermal-hydraulic behavior. A review of such codes and capabilities is presented in Ref [11]. In the application of STARRS, one can, in

principle, combine STARRS to desired thermal-hydraulic codes in order to provide appropriate boundary conditions.

The STARRS computer code was exercised extensively, validated using available experimental data, and then applied to plant conditions. The following provides an overview of the code applications, and has been divided into three specific areas:

Sensitivity calculations: STARRS was exercised extensively to identify the involved competing mechanisms, and to determine the dependence of the key parameters (for example, the decontamination factors, and radioactivity release rate) on operating and thermal-hydraulic parameters, such as:

1. iodine partition coefficient,
2. two-phase swell level,
3. location of break relative to swell level,
4. primary and secondary-side iodine concentration, and
5. initial size distribution of flashing aerosols.

Both transient and steady state calculations were performed.

Code validations: the STARRS transport and retention model was validated using the Model Boiler 2 (MB-2) experimental data [12, 13]. Both the phase II radioactivity simulation experiments (using KOH in the primary and LiOH in the secondary-sides as trace elements) and the dry secondary side experiments were utilized. More than 12 steady-state and 5 transient validation cases were performed.

Plant application: STARRS was applied to a U-tube steam generator in order to predict the release rates for a plant geometry operating at prototypical conditions. Sensitivity calculations were also performed to assess plant releases as a function of iodine partition coefficient and the break location relative to the swell level. Furthermore, scaling calculations were performed to assess differences in iodine releases that may occur in both the MB-2 and actual plant steam generators.

The details of validation and application calculations can be found in [14]. Sample results are presented below.

Figures 4 and 5 depict the calculated differential decontamination factors as functions of the iodine mass fraction on the primary and secondary sides, respectively. The results indicate the following: (1) higher DF's are obtained for higher partition coefficient, (2) there is considerable removal between the swell level and the steam generator exit, (3) the DF increases as the primary-side concentration increases (for a given secondary-side concentration), and (4) the DF decreases as the secondary-side iodine mass fraction increases (for a given primary-side concentration). The above results imply that the iodine vapor contribution to the total release can be significant and strongly depends on the partition coefficient.

Selected validation results are shown in Figures 6 through 10 for Tests T-1970 and T-2053. The conditions for Test T-1970 were as follows: primary-side pressure = 3.84 MPa, primary-side temperature at the break = 493K, secondary-side presure = 2.0 MPa, boil-off rate = 0.195 kg/s, secondary-side liquid mass = 600 kg, downcomer water level = 11.3 m, break height above lower tube sheet = 0.15 m, breakflow rate = 0.17 kg/s, primary-side K mass fraction = 40 ppm, and initial K mass fraction on the secondary side = $\tilde{}10^{-3}$ ppm. Figure (6) shows a comparison between the measured and predicted K mass fraction in the secondary-side liquid. As can be seen, the agreement is very good, indicating that the bulk of the K which flows through the break remains with the secondary-side liquid. Figure (7) shows a comparison between the measured and predicted K mass fraction at the steam generator exit. The predictions are shown for two values of the KOH partition coefficient, which are believed to be within the uncertainty bound. As can be seen, the predictions and data are in good agreement. The initial high value of K data is due to the residuals from the previous test; model predictions are based on zero initial concentration in the system at the beginning of the simulation.

Figure (8) shows that the escaped K from the system is comprised mainly of vapor iodine and of liquid aerosols entrained at the swell level. Since the break is submerged and its location is close to the lower tube sheet, the aerosols formed by flashing of the break flow (primary-side fluid) are removed very efficiently and neglible amounts escape from the system.

Figure (9) shows the predicted and measured K concentrations at the SG exit for Test T-2053. The predictions are shown for 3 values of swell level and two values of partition coefficient. The conditions for T-2053 were as follows: primary-side pressure = 12.75 MPa, primary side temperature = 578K, secondary-side pressure = 7.44 MPa, boil-off rate = 0.86 kg/s, liquid mass in the secondary-side

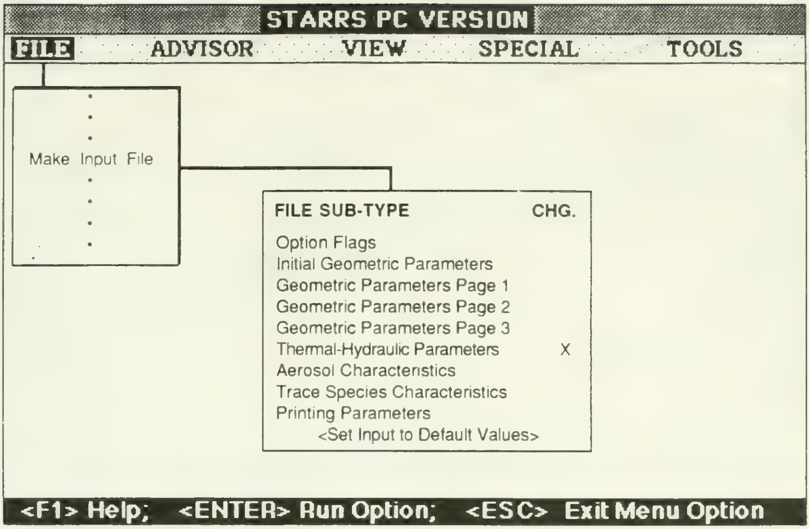


FIGURE 3 SAMPLE OF STARRS PULL-DOWN MENU SYSTEM

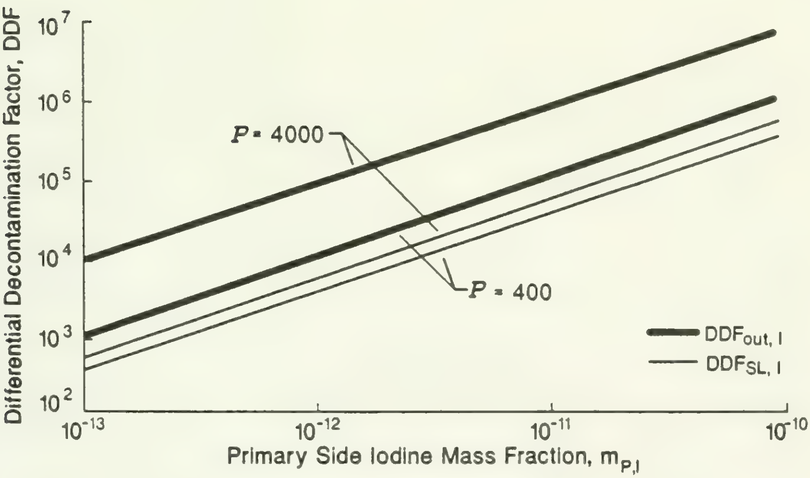


FIGURE 4 EFFECT OF PRIMARY-SIDE IODINE MASS FRACTION ON DF

$M_{S,I} = 10^{-16}$, $Z_{SL} = 13.66$ m, $Z_B = 6$ m

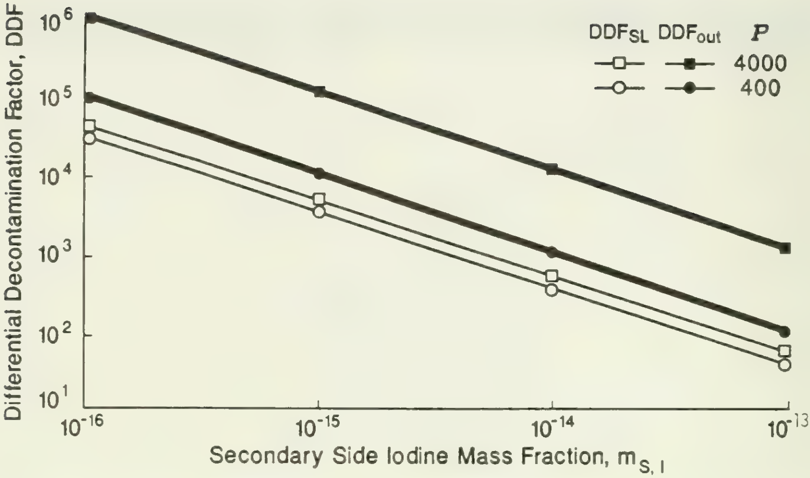


FIGURE 5 EFFECT OF SECONDARY-SIDE IODINE MASS FRACTION ON DF

$M_{P,I} = 10^{-11}$, $Z_{SL} = 13.66$ m, $Z_B = 6$ m

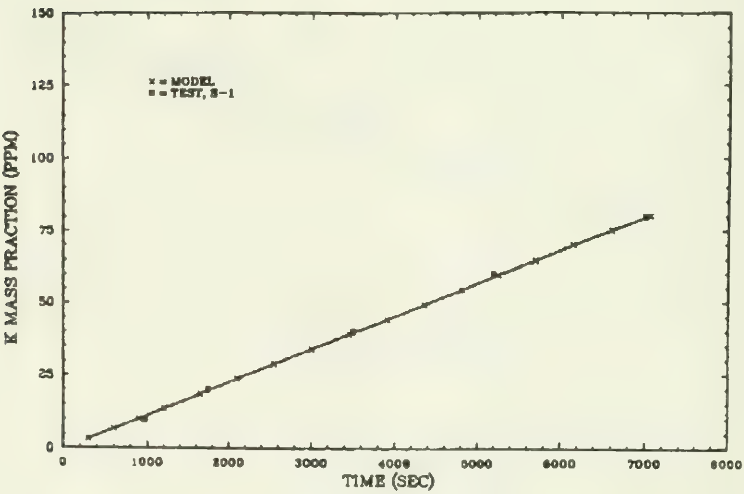


FIGURE 6 K MASS FRACTION IN SECONDARY WATER IN TEST 1970 OF MB-2

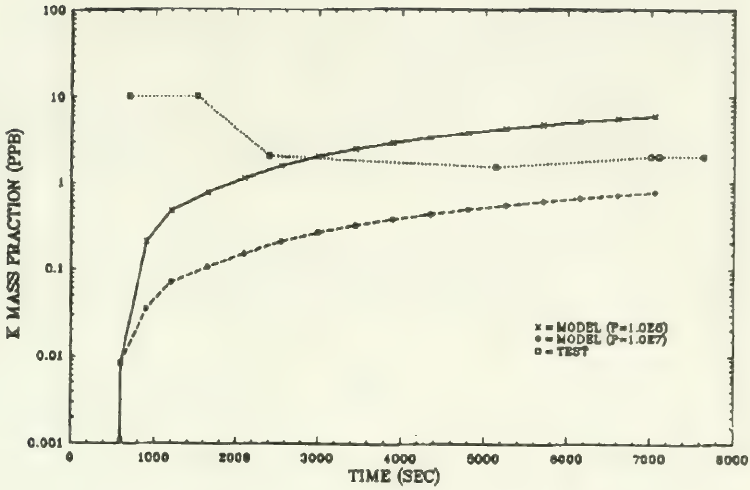


FIGURE 7 K MASS FRACTION AT EXIT IN TEST 1970 OF MB-2

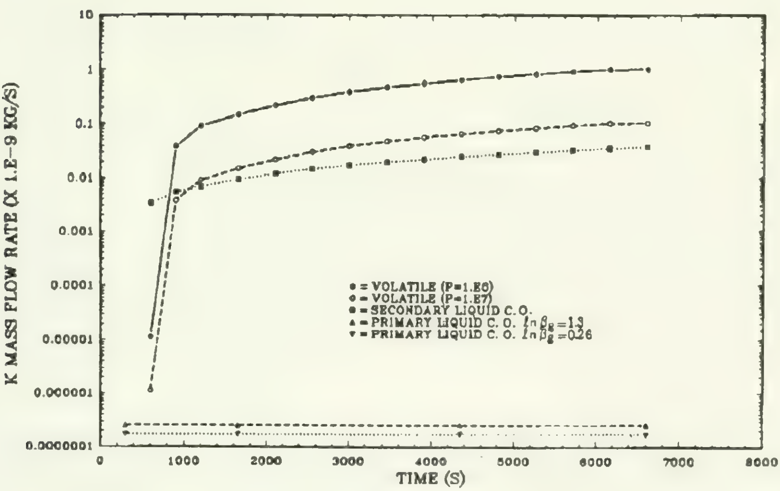


FIGURE 8 COMPONENTS OF K MASS FLOW RATE AT EXIT (TEST T-1970 MB-2)

= 130 kg, downcomer water level = 4.06 m, break height above lower tube sheet = 7.11 m, break flow rate = 0.23 kg/s, primary side K mass fraction = 52 ppm, initial secondary-side K mass fraction in the liquid = 150 ppm. For this test the break is very close to the swell level, and as can be seen in Figure (9), the predictions are markedly affected by relatively small variations in the swell level. This is because the entrained iodine in the rising bubbles does not reach equilibrium with the secondary-side liquid before the rising bubble reaches the swell level due to the short rise distance.

Figure (10) shows the make-up of the exit stream. For this case, since the break is close to the swell level, the contribution of the flashing aerosols is as significant as the contribution of the vapor iodine.

The STARRS code application was extended to a Model F Westinghouse steam generator, where detailed comparisons between the Model-F and the scaled MB-2 steam generator were performed, in order to assess differences (if any) in the secondary-side retention. Additionally, decontamination factors and radioactivity rates were calculated as a function of operating conditions, and iodine partition coefficients. Selected results are shown for the following conditions: primary-side pressure = 12.75 MPa, primary-side coolant temperature at the break = 578 K, secondary-side pressure = 7.44 MPa, secondary-side liquid mass = 7.14×10^4 kg, secondary-side boil-off rate = 113.9 kg/s, break height above lower tube sheet = 8.8 m, break flow rate = 19.8 kg/s, primary-side iodine mass fraction = 2.02×10^{-13} (corresponds to 25 nCi/g), and swell level height = 13.74 m.

Figures 11, 12, 13, and 14 show respectively the following: the mass fraction of iodine in the secondary liquid, the differential and integral DF's, the rate of radioactivity release, and cumulative radioactivity release. As can be seen, the results are sensitive to the partition coefficient value. The DF's depend strongly on the partition coefficient since the contribution of vapor iodine is significant. The decrease of DF with time is due to the build-up of iodine on the secondary-side.

SUMMARY

An interactive PC version of the STARRS computer code has been developed to quantify the magnitude of radioactivity release during an SGTR event. This user-friendly code provides the utility engineer with appropriate guidelines on how to apply the code and analyze the results. It also includes a memory-resident graphics capability.

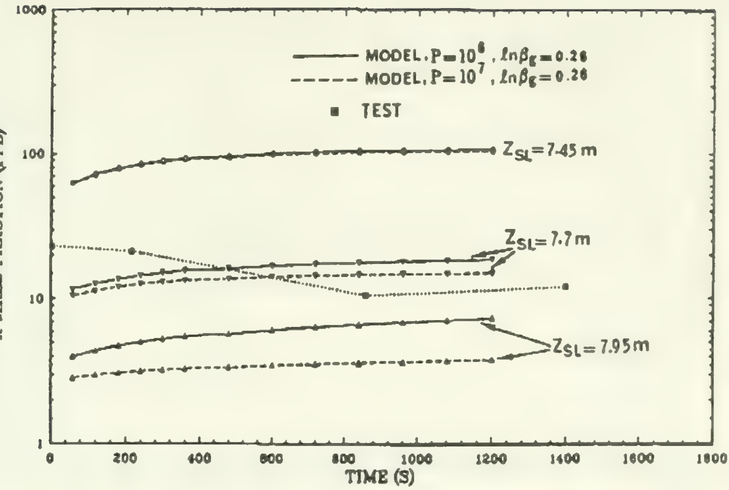


FIGURE 9 K MASS FRACTION AT EXIT IN TEST T-2053 OF MB-2

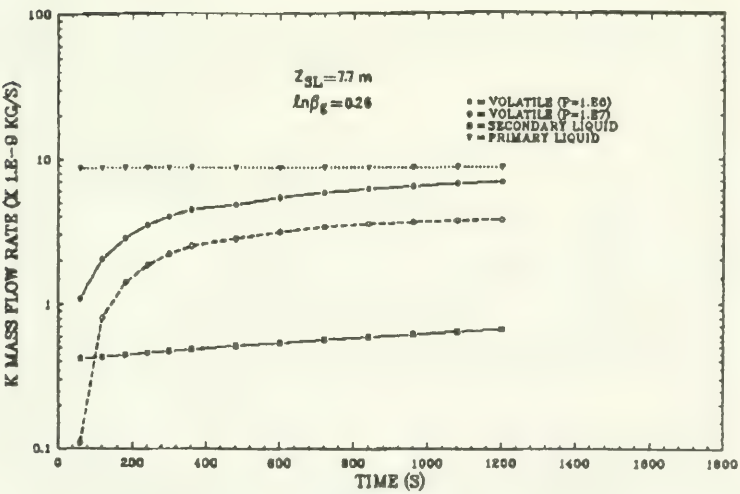


FIGURE 10 COMPONENTS OF K FLOW RATE AT EXIT (TEST T-2053 MB-2)

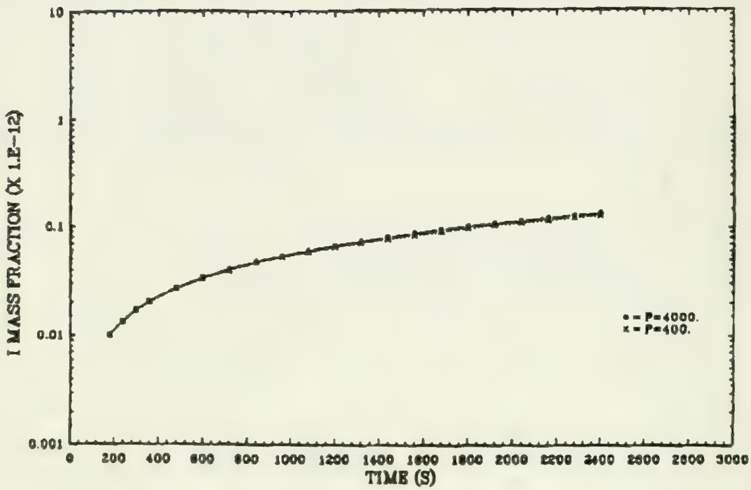


FIGURE 11 IODINE MASS FRACTION IN PROTOTYPE SECONDARY-SIDE LIQUID

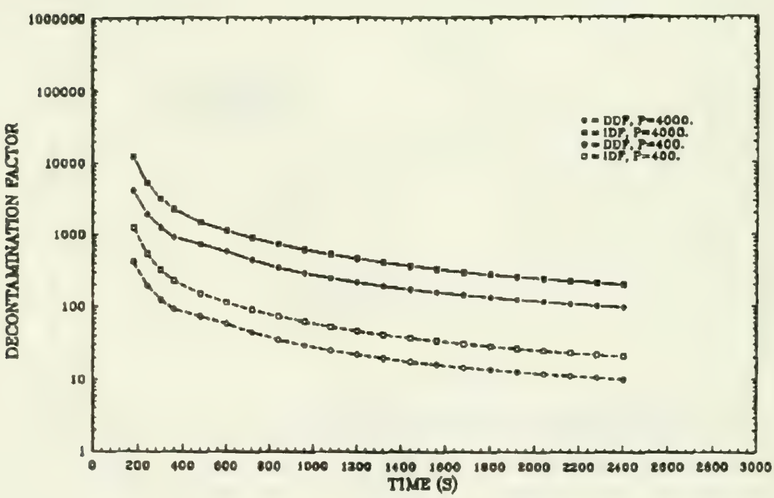


FIGURE 12 DECONTAMINATION FACTORS IN PROTOTYPE

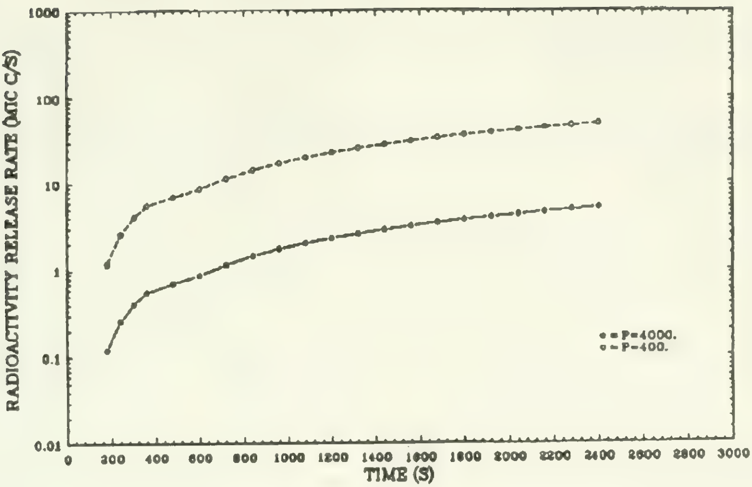


FIGURE 13 RATE OF RADIOACTIVITY RELEASE IN PROTOTYPE

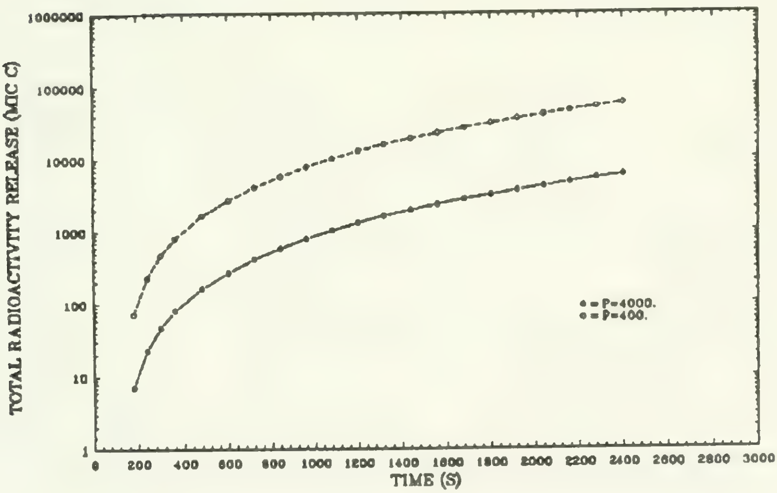


FIGURE 14 CUMULATIVE RADIOACTIVITY RELEASE IN PROTOTYPE

STARRS was validated using available experimental data and was applied to prototypical plant conditions. Extensive sensitivity calculations were performed.

REFERENCES

1. J. Hopenfeld. "Radioactive Transport Following SGTR," NRC NUREG-1108 Report, March, 1985.

2. D. G. Cain. "Small Artificial Reasoning Tool Kit," EPRI Report NP-5645M, July 1988.

3. S. M. Ghiaasiaan and A. T. Wassel. "Secondary-side Transport and Retention of Radioactive Species (STARRS) Computer Code; Vol I, Transport and Retention Analysis," EPRI Report to be published.

4. R. Brown and J. L. York. "Sprays Formed by Flashing Liquid Jets," AIChE Journal, 8 pp. 149-153, 1962.

5. J. O. Hinze. "Fundamentals of Hydrodynamic Mechanism of Splitting in Disperson Processes," AIChE J., 1.pp 289-295, 1955.

6. J. H. Lienhard and J. B. Day. "The Breakup of Superheated Liquid Jets," J. Basic Engineering, 88, pp. 515-522, 1970.

7. T. H. Lehrer. "On Bubble and Drop Formation and Breakup," Israel J. of Technology, 13, pp. 246-252, 1975.

8. T. Z. Harmathy. "Velocity of Large Drops and Bubbles in Media of Infinite or Restricted Extent," AIChE Journal, 6, p. 281, 1960.

9. I. Ktaoka and M. Ishii. "Mechanistic Modeling of Pool Entrainment Phenomenon," Int. J. Heat and Mass Transfer, 27, pp. 2000-2014, 1984.

10. M. Y. Young, K. Takeuchi, O. J. Mendler and G. W. Hopkins. "Prototypical Steam Generator Transient Testing Program: Test Plan/Scaling Analysis," EPRI NP-3494, 1984.

11. S. P. Kalra. "Modelling Transients Steam Generator Units," Nuclear Safety Journal, Vol. 25, No. 1, pp. 33-52, 1984.

12. K. Garbett, et al. "Coincident Steam Generator Tube Rupture and Stuck-Open Safety Relief Valve Carry-over Tests," EPRI NP-4787, NUREG/CR-4752 2, 1987.

13. H. A. Morewitz, et al. "Test Report: MB-2 Aerosol Attenuation Tests," Part of EPRI Report NP4787, 1987.

14. S. M. Ghiaasiaan and A. T. Wassel. "Secondary-side Transport and Retention of Radioactive Species (STARRS) Computer Code: Vol IV, Applications," EPRI Report to be published.

Evaluating Plant Modifications against Industry Operating Experience—The Industry Experience Advisor

JERRY D. SWISSHELM and JAMES H. RILEY
Systems Engineering Division
Impell Corporation
333 Research Court
Technology Park/Atlanta
Norcross, Georgia 30092, USA

LARRY F. PABST
Special Projects Manager
Nuclear Engineering
Florida Power & Light Company
P.O. Box 14000
Juno Beach, Florida 33408, USA

ABSTRACT

To avoid repeating problems of the past, the NRC requires utilities to review plant design and future modifications against industry operating experience. However, so much industry experience information exists that it is virtually impossible to consistently identify industry experience related to a given modification. To overcome this difficulty, Florida Power & Light and Impell Corporation developed a categorized list of root causes of industry events called the Operating Experience Feedback Checklist. Root causes were determined by a senior systems engineer's in depth review of each industry experience document. To subsequently improve consistency and reliability in use of this checklist, Florida Power & Light and Impell Corporation developed a prototype expert system, the Industry Experience Advisor, that would assist an engineer in identifying root causes or lessons learned in the checklist that are applicable to a given plant modification.

INTRODUCTION

In 1980, as a consequence of the lessons learned from the Three Mile Island accident, the NRC issued requirements for nuclear power plant licensees to implement an operating experience assessment function.

This action stressed the importance of operating experience to the improvement of overall reactor safety. While operating experience feedback is routinely reviewed against the existing plant design, there has not been any efficient and systematic means of reviewing pending design changes against previous industry experience.

As part of a Plant Change Risk Assessment Program at Plant Turkey Point (PTP) (Reference 1), Florida Power & Light Company (FP&L) sought an innovative method of reviewing existing design change packages against previous industry experience. FP&L sought a means by which Plant Change/Modification packages (PCM's) could be efficiently reviewed against important operating experience, taking into account the root causes or lessons learned from past industry events. Existing methods, such as abstract and keyword search, were too time consuming, and did not provide the required level of information for this project. FP&L came to Impell Corporation with this problem, because of Impell's broad nuclear industry engineering experience and advanced software techniques.

FP&L and Impell developed a two-fold solution to this problem. First, operating experience documents would be reviewed in depth by senior systems engineers to determine the root cause or lesson learned of each industry event, and the root causes categorized into a checklist to allow for manual review of existing PCM's. Second, an expert system would be built that would include the checklist data, and assist a plant engineer in quickly reviewing design changes against previous operating experience.

This paper discusses the evaluation and organization of the operating experience data, and the development of the expert system prototype used to access this information.

Work on this project began in December, 1987 and was completed in April, 1989.

REVIEW AND ORGANIZATION OF INDUSTRY OPERATING EXPERIENCE INFORMATION

The first step in developing a process to review and organize industry operating experience is to identify sources which contain the most useful Operating Experience Feedback (OEF). The best sources should contain information which has been pre-screened both to remove unimportant data and to present events which have either industry wide significance or are specifically related to Plant Turkey Point. During the project, optimum use was made of existing operating experience feedback screening efforts (e.g. NRC, INPO) to ensure that all the important generic concerns that might be related to a design change were reviewed with a minimum amount of duplicated effort. The intent of the project was to review design changes against <u>important</u> industry operating experience, not <u>all</u> industry operating experience.

Sources of Operating Experience

There is considerable information available on industry experience ranging in detail from individual Licensee Event Reports (LERs) to the NRC's Office for Analysis and Evaluation of Operational Data's (AEOD) quarterly report to Congress on abnormal occurrences. Optimally the experience documentation reviewed for this project should have screened large amounts of experience data into important generic issues while still maintaining sufficient detail to allow specific events to be evaluated.

Our project evaluated the following as potential sources of operating experience feedback documentation.

- NRC
- Institute of Nuclear Power Operations (INPO)
- Turkey Point/FPL Internal Experience
- Vendor/AE Technical Information

From these potential sources, we selected the following documents for review in this project (approximately 2000 documents issued through 1987).

- NRC IE Notices, Circulars, and Bulletins
- INPO SERs and SOERs
- Turkey Point LERs

We felt that these documents represented the best sources of industry event information for this project for the following reasons.

- They dealt primarily with design issues, as opposed to regulatory issues.
- They presented specific information about industry events, including causal evaluations.
- They covered the complete spectrum of operating experience, as opposed to examples of selected events.

Operating Experience Review and Root Cause Determination

The information obtained from the selected documents was entered into a dBASE III formatted computer data base which was used to create a summary checklist of operating experience. Use of the computer also allowed later sorting of the information using several different

parameters and facilitated the development of the expert system. The data recorded for each event included: document title and number, nuclear plant(s) at which the event occurred, affected systems and components, reference documents, and, most importantly, the "lessons learned" or the "root causes" of the event described.

This root cause determination was critical to the performance of the design change review. It required that the event described in the OEF document be subjected to a thorough engineering evaluation by a senior systems engineer to identify the basic problem, or problems, involved. The cause of the problem or the lesson learned from the event had to be accurately and concisely stated. A significant level of systems experience and effort was required to ensure that the lesson learned was general enough to allow interpretation and application to design changes involving different systems, components, or structures. For example, a lesson learned from a misapplication of a type of relay in a BWR Reactor Protection System circuit could also potentially apply to a design change involving a similar type of relay in a PWR system. At the same time the root cause must be specific enough to avoid restating "motherhood" design principles such as "check valves leak" which convey no useful information to the plant engineer.

Operating Experience Feedback Checklist. The Operating Experience Feedback (OEF) Checklist (sample provided in Table 1) is a listing of all the "root causes" or "lessons learned" relating to the design change process at Turkey Point, that were developed during the review of operating experience feedback documents. The checklist is organized by categorizing root causes under headings and sub-headings, which we termed "Tables" and "Topics", respectively. The two tables included in the checklist are:

- Table 1 -Topics Related to the Physical Characteristics of Design Changes

- Table 2 -Topics Related to Administrative Aspects of the Design Change Process

A third table, entitled "Topics Unrelated to the Design Change Process at Turkey Point", contains all the root causes from the review of operating experience feedback documents that fall outside of the scope of this project. This table and the root causes associated with it are not included in the checklist.

Each of the tables is divided into topics as shown in Tables 2 and 3. It is readily apparent that these topics are still so broad that further subdivision is necessary in order to be able to efficiently locate root causes related to a particular design change. Therefore each of the topics is subdivided into subtopics such as the "system" heading shown in Table 1.

OEF Checklist Use. The OEF Checklist was organized so that it could be used directly by the design engineer or the design change reviewer. The basic entry point into the checklist is at the topic level. In order to determine which topics are applicable to a design change, answers to the following questions are required.

1. What systems are physically affected by the design change?

2. What component types (e.g. valves, pumps, relays) are affected by the design change?

3. Are component or piping supports affected?

4. Are buildings or other structures being modified?

5. Is the physical or functional configuration of the system affected by the design change?

6. Does the design change affect safety related systems and components, or equipment qualification?

7. Are the vendors known for the components affected by the change? If so, list them.

8. Is information known about the water chemistry or materials used in the systems or components affected by the design change?

9. Does the design change affect security systems or does it, in some way, concern the physical security of the site?

These questions and the OEF Checklist itself were the bases upon which the expert system described in the next section was developed.

EXPERT SYSTEM DEVELOPMENT

To avoid the cost of codifying the entire checklist (over 800 root cause statements) during prototype expert system development, a representative cross-section of the OEF Checklist was chosen for entry into the knowledgebase. The chosen cross-section represents all root causes relating to 1987 industry documents. This cross-section contained 75 root causes (slightly less than 10%), with representation in all but one of the Topics (Table 1 Topic 9 - Vendor Specific).

The Expert for the expert system development was Mr. James Riley, an Impell Supervising Engineer with over 16 years of engineering experience, over 12 of which were in the nuclear industry. Mr. Riley

was responsible for the OEF Checklist design, and review of PCM's using the Checklist.

Mr. Riley's knowledge was important to provide a firm foundation for the checklist knowledge, and to provide the knowledge that would bridge the gap between the checklist and plant modifications. Mr. Riley and his project team were observed and interviewed during some of the checklist preparation. Expert interviews were conducted to research the checklist basis. Mr. Riley and some of his staff were interviewed and observed while they were using the checklist to review PCM packages.

<u>Checklist-based PCM Review</u>

The observed activities in the checklist-based PCM review were to:

1. Describe the plant modification by reading the PCM cover sheet and important pages.

2. Compare the descriptive parameters with the Table/Topic/Subtopic groupings in the checklist to determine which root causes to read in detail.

3. Read the root causes in detail to determine if the industry documents associated with the root cause are potentially applicable.

4. If necessary, review the PCM in more depth for comparison against conditions implied by the root cause statement.

From observation of the Expert and project staff performing checklist-based PCM reviews, it became apparent that they relied upon knowledge of the checklist basis and industry documents themselves to supplement what was provided in the root cause statements. To capture this knowledge without extensive interviewing and recording of many PCM reviews, we decided to have the Expert expand information on the checklist subtopics and root cause statements. To assist in understanding the root cause statements, the Expert developed statements or questions that describe what conditions, in addition to the Subtopic heading, would be necessary for the root cause to be applicable. The Expert revealed that many times compromises were made to determine under which subtopic a given root cause should be located. This was done to maintain the linear nature of the checklist, and limit its length. To recover these lost relationships, the Expert was also asked to add to each root cause cross-references to other subtopics where appropriate.

The Expert's answers revealed a high level division of the checklist that is not represented in its table/topic structure. This division was according to plant change type, which was narrowed down to the following categories:

1. mechanical

2. electrical

3. instrumentation and control

4. civil/structural

All root causes were classified to belong to one or more of these types of changes, in addition to the table/topic structure.

Knowledge Representation

Impell chose the NEXPERT Object expert system shell from Neuron Data for this project. This choice was based on combined needs for an object-oriented approach towards storing root cause information and classifications, a flexible rule-based system for evaluating design change applicability of a root cause, and database interface for accessing the root cause information. The NEXPERT Object Developer's Environment (Reference 2) was used for entry, review, and testing of the knowledgebase, and the NEXPERT Object Runtime Environment (NORT) (Reference 3) was used to build the user interface and help facilities. Where NEXPERT did not provide the desired capability or control structure, 'C' code was written and linked through the NEXPERT Callable Interface (Reference 4).

The representation scheme was chosen to limit the amount of information presented to the user to that which was determined applicable. The representation scheme was also designed to minimize execution speed, and provide ease of maintenance. Rules were written to only represent generic or procedural information. Classes and objects were used to represent the specific data. Generic rules would then be reused to operate on specific data, minimizing the rulebase size and the execution speed.

Class Structure. The OEF Checklist was composed of a collection of objects - the root causes, belonging to groupings arranged in a hierarchical order. However, the Expert identified many possible cross-memberships in the checklist tree structure, and identified new relationships not a part of the current tree (the plant modification types described above). This was easily implemented in the knowledgebase by representing the topics, subtopics, and change types in a multi-linked class structure, with attribute inheritance. The

root causes were represented as objects belonging to many different classes at different levels of the class structure.

For example, Figure 1 shows an object (RC_11) that is a root cause (belongs to the class of root_causes). It is simultaneously a member of the checklist subtopics logic under table1_topic2, and valves under table1_topic4. It is also a member of the change types mechanical and instrumentation_control. This is a higher level representation than was available in the linear checklist.

The change types (mechanical, electrical, instrumentation_control, civil_structural) were designed as classes with root cause objects belonging to them. These classes are used to limit the focus of the user to only those root causes on the checklist that belong to the change types selected. The topics and subtopics were also designed as classes with root cause objects belonging to them. These classes are used to further limit the focus of the user.

<u>Root Cause Conditions.</u> In addition to their memberships, root causes contained conditions determining when the root cause was applicable. After a root cause was selected as potentially applicable based on its class memberships, this information was used to determine if it was actually applicable. Figure 2 shows a root cause (RC_12) containing three conditions (piping_geometry_changes, piping_material_changes, fluid_condition_changes). If any of these conditions is present (piping_geometry_changes or piping_material_changes or fluid_condition_changes), then the root cause is applicable.

<u>Rule Structure.</u> Many root causes had multiple conditions that proved their applicability. Almost all of the multiple-condition root causes determined by the Expert required an 'or' operator to prove applicability - i.e. if either condition A <u>or</u> condition B was present, then the root cause was applicable. NEXPERT does not provide an 'or' operator in its rule syntax. Multiple rules could have been written that point to the same root cause, thereby performing the 'or' operation. This would have required root cause specific rules. This approach was not used, because of the required rulebase size for the 800+ root causes of the full checklist, and the difficulty in adding and maintaining rules. Instead, an external routine was written in 'C' to perform the 'or' operation on condition subobjects belonging to the root cause objects. In other words, the conditions were represented as subobjects that belonged to one or many root cause objects (root causes may share the same condition), where the presence of any of the conditions proved the applicability of the root cause.

Where an 'and' operator was required, a new condition was defined with a rule proving it - i.e. if condition A <u>and</u> condition B were present, then condition A_and_B was present (condition 'A_and_B' would be the condition belonging to the root cause). This did not re-

quire root cause specific rules - only condition specific rules. Conditions were intended to be generic - i.e. available for use by any root causes that needed them.

A condition was proved present by either asking the user, or by searching the rulebase for rules that might prove it present from other conditions, and evaluating the rules. For example, a root cause may have the condition 'increases_in_flooding_probability' that must be present to prove the root cause applicable. Figure 3 shows this condition using three rules to prove that it is present. These rules use the presence of other conditions in their proof. The rules are searched non-exhaustively, i.e. as soon as one is found true, then the required condition is proved present.

Control and Program Flow

The execution of an advisory session follows these steps.

1. Determine the type(s) of plant changes present in this modification, and extract from the database root causes that belong to the selected change types. Label each selected root cause with the change type that caused its selection for user explanation.

2. Next, help the user generally describe the plant modification by presenting appropriate topics and subtopics to choose from. Select only root causes that belong to the chosen topics and subtopics, eliminating the others from further consideration. Label each with the topics and/or subtopics that cause the selection for user explanation.

3. Show the resulting list of root causes, and the labels explaining why they were selected. Allow the user to delete any root causes that can be justified as not applicable, where the user enters an explanation for the deletion.

4. Evaluate the remaining root causes individually by evaluation of the plant change conditions that must be present for the root cause to be applicable.

Order of Processing. First the change types (mechanical, electrical, instrumentation_control, civil_structural) describing this PCM are selected. Then affected systems and affected components are selected. Then the remaining affected checklist tables, topics, and subtopics are selected. This results in a list of selected root causes based on the class memberships. The next step is to evaluate the selected root causes individually by examining the conditions as-

sociated with each root cause. The user is asked about the condition. If the user answers NOTKNOWN (he/she doesn't know or the information is not available), then the rulebase is searched to try to prove the condition by backward chaining. If this fails, then a default (usually condition present is TRUE) is assumed. This process labels a root cause as truly applicable. When all selected root causes have been evaluated, then the PCM evaluation is complete, and reports describing the applicable root causes are generated.

User Prompts. Asking the right question proved to be an important part of the knowledge representation. NEXPERT's capability to provide a specific prompt for each condition under evaluation was important. Through careful definition of the condition name, default prompts could be inherited and used. But where the default prompt did not make sense, or where a more detailed question would be valuable, a specific prompt was entered for that condition.

User Help. The availability of help was important to the use of the system. This capability was easily implemented using NORT's context sensitive help facility. A single line of help was always available, and more detailed help in the form of a text file review was available at the press of a function key. The creation of help files is an easy but time-consuming task. Therefore, only representative help files were created with the prototype for demonstration, realizing that many more could be added later.

Figure 4 presents an example screen from the program showing the table and topic selections presented to the user, and the context-sensitive one-line help at the bottom of the screen.

Knowledge Expansion and Maintenance

The prototype was designed to allow easy expansion and maintenance. This is key to successful expert system software methodology. New ideas, knowledge, and approaches can be quickly implemented and tested.

To add industry experience in the form of a new root cause, the database is easily updated using the dBASE III database manager. Table, topic, and subtopic memberships are defined next and entered into the knowledgebase by creating a new root cause object with appropriate class memberships. Conditions required for design change applicability are created as subobjects of the root cause object. This is all performed using the NEXPERT Developer's Environment Object Editor. This object-oriented approach requires no rule additions or modifications, greatly reducing the chance of error introduction.

The rules developed to prove the presence of a condition are root cause independent. They can be expanded for more in depth user assistance without affecting the industry experience root cause data.

RESULTS

The OEF Checklist demonstrated an efficient method of presenting industry operating experience information about the root causes or lessons learned from an industry event. The OEF Checklist was used successfully by its developers to review over 1700 PTP PCM's against previous operating experience, in a fraction of the time it would have taken using keyword search, abstract, and full document review.

The Industry Experience Advisor prototype expert system demonstrated the ability to capture expert knowledge about the OEF Checklist, and provide it in a standardized format as an expert system. An expert system knowledgebase was built that included the OEF Checklist root causes relating to 1987 industry documents.

Expert knowledge used to develop the OEF Checklist was recovered and implemented in the system. This included having an expert reclassify the root causes under multiple checklist topics, provide a detail explanation of the meanings of the checklist topics and subtopics, and describe conditions for each root cause that determine its applicability to a plant change.

A knowledge representation scheme was developed that limited the amount of information presented to the user to that which is applicable to the plant change under investigation. The knowledge representation was designed to be easily expandable and maintainable. User prompts and response methods were investigated, and a workable interface implemented. The knowledge was integrated into an expert system shell that provided powerful development and testing capabilities. The resulting system was delivered for development testing.

RECOMMENDATIONS

The Industry Experience Advisor Prototype is intended to be a first-order prototype. Its purpose is to test techniques and knowledge representations for developing an eventual production system. The limited testing performed during the development indicated a production system is achievable. The following further steps towards this goal are being pursued.

Development Testing

More extensive testing needs to be performed and documented. The tests should compare using the prototype against using the manual 1987 checklist. The tests should consider ease of use, accuracy, and documentation requirements.

Enhancements

Improvements can be made to the prototype that will allow more meaningful productivity comparisons. These would lead to a Fieldable Prototype to be tested under actual field or work conditions.

Expand the System for all Root Causes. Post 1987 documents need to be reviewed, and root causes extracted. All of the remaining root causes (pre- and post-1987) need to be added to the system. Expert knowledge should be acquired concerning topic/subtopic memberships, and appropriate conditions for each root cause.

Optimize the Root Cause Conditions. The condition list should be reviewed, and related conditions linked. Wherever conditions can be proven by other conditions, rules should be added, and a forward-chaining mechanism implemented to prove conditions before prompting the user.

Field Test the System. The resulting Fieldable Prototype should then be tested in a production environment. Comparisons should again be made against manual methods, except now actual productivity comparisons should be made. As before, any program errors or user recommendations should be written down.

Develop the Production System. The entire system design should be reviewed against user experiences, requirements and recommendations. The program should be cycled through standard software development procedures, with formal specifications and design review. Program control and maintenance procedures should be established. The program should be redesigned for optimal performance on the delivery hardware, and recoded as necessary. Complete QA verification testing should be performed for use in nuclear safety-related work.

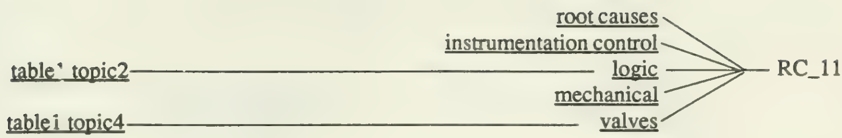


Figure 1. Example Root Cause Memberships

Figure 2. Example Root Cause Memberships and Conditions

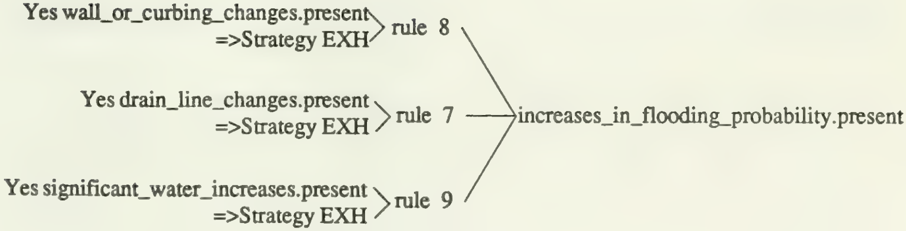


Figure 3. Example Rule Network Proving a Condition

****

* Industry Experience Advisor Prototype Version 1.1 *

****

Table 1 - Physical Aspects of Design Changes

Topic 2 - System Configuration Select the area(s) of System Configuration that are affected by this change:	electrical design **electrical layout** instrument design instrument layout logic mechanical design mechanical layout miniflow piping design piping layout
Topic 3 - Coolant Boundary Integrity Select if you know the water chemistry or materials used in the systems or components affected by this change:	chemistry materials

physical configuration or location of electrical equipment

Figure 4. Example Screen Showing Context Sensitive Help

TABLE 1

EXAMPLE PAGE FROM OEF CHECKLIST

OEF CHECKLIST
1987 DOCUMENTS ONLY
TABLE 1-PHYSICAL ASPECTS OF DESIGN CHANGES

TOPIC 1-SYSTEM SPECIFIC

DOCUMENT NUMBER	SYSTEM	ROOTCAUSE	RELATED DOCUMENTS
SER 005-87	COND	MODS THAT AFFECT CONDENSER VACUUM MAY CHANGE FATIGUE LOADING ON TURBINE BLADES CAUSING FAILURE	
250-87-020	COOLING WATER	USE OF A CLOSED LOOP SALT WATER COOLING SYSTEM CAUSES HIGHER CARBONATE CONCENTRATIONS AND INCREASED FOULING OF HEAT EXCHANGER SURFACES	
SER 030-87	EDG	DG CONTROL SYSTEM INTERLOCKS PREVENT EXCITER FIELD FLASHING IF START SIGNAL IS RECEIVED IMMEDIATELY AFTER A DIESEL SHUTDOWN	IEN 83-17,IEN 86-73,SOER 86-073
250-87-007	EDG	WATER ACCUMULATION IN DIESEL FUEL OIL CAUSES FUEL DEGRADATION	
SER 006-87	FP	CO2 FIRE PROTECTION SYSTEMS SHOULD CONTAIN ODORIZERS FOR PERSONNEL PROTECTION	
SER 029-87	FP	FP SYSTEM SPRAY NOZZLE ORIENTATION IMPACTS EQUIPMENT (TRANSFORMER) OPERABILITY/SUPPRESSION CAPABILITY	
IEN 87-020	GAS/AIR	VALVES WITH CONVENTIONAL STEM PACKING ALLOW EXCESSIVE LEAKAGE IN GAS SYSTEMS.	
IEN 87-020	GAS/AIR	ROUTING OF COMBUSTIBLE AND TOXIC GAS LINES MUST CONSIDER EFFECTS OF LEAKAGE.	
IEN 87-020 PG2	HVAC	HVAC FLOW REQUIREMENTS MUST CONSIDER POSSIBLE COMBUSTIBLE & TOXIC GAS LINE LEAKAGE.	
IEN 87-013	SFSS	INADEQUATE LEAK DETECTION SYSTEM FOR REDUNDANT PNEUMATIC SEALS.	

TABLE 2

OEF CHECKLIST TABLE 1 TOPICS
Topics Related to the Physical Characteristics of Design Changes

Topic	Explanation
1. System Specific	Industry experience issues related to a specific system are listed here by system.
2. System Configuration	Issues related in a generic sense to system design provisions, not associated with a specific system.
3. Coolant Boundary Integrity	Issues related to choice of materials or water chemistry.
4. Component Specific	Issues related to individual components listed by component type (pump, valve, relay, etc.).
5. Component Support	Issues related to pipe and conduit supports, snubbers, and the civil engineering aspects of component support.
6. Structures	Architectural related issues.
7. Human Factors	Issues related to the human element of design.
8. Personnel Protection	Issues related to the physical design aspects of ALARA and occupational safety.
9. Vendor Specific	Issues documenting where a specific vendor has been found to supply unacceptable materials.

TABLE 3

OEF CHECKLIST TABLE 2 TOPICS
Topics Related to the Administrative
Aspects of the Design Change Process

Topic	Explanation
1. Calculation/Analysis	Issues related to calculational accuracy or methodology.
2. Q List	Issues related to the safety classification of systems and components.
3. Personnel Protection	Issues related to ALARA and occupational safety concerns arising during the design change process.
4. EQ and Seismic	Issues related to design for accident conditions
5. Environmental Design	Issues related to normal environmental design conditions.
6. Plant Security	Issues related to maintaining plant security provisions, as opposed to changes to the security systems (security system design changes were not reviewed during this project since they involve safeguards information).

REFERENCES

1. Impell Corporation, "FP&L Plant Turkey Point PCM Screening Project Final Report," Report Number 03-1050-1154, Revision 0 , June 30, 1988

2. Neuron Data, Inc., "NEXPERT Object Fundamentals", Version 1.1, IBM-AT Version

3. Neuron Data, Inc., "NEXPERT Object Runtime (NORT) User's Manual", Version 1.1, IBM AT Version

4. Neuron Data, Inc., "NEXPERT Object Callable Interface Manual", Version 1.1, IBM AT Version

Expert Systems Use in Present and Future CANDU Nuclear Power Supply Systems

L. R. LUPTON, R. A. J. BASSO, and L. L. ANDERSON
Atomic Energy of Canada, Research Company
Instrumentation & Control Branch
Chalk River Nuclear Laboratories
Chalk River, Ontario, Canada K0J 1J0

J. W. D. ANDERSON
Atomic Energy of Canada Limited, CANDU Operations
Licensing and Risk Branch
Sheridan Park Research Community
Mississauga, Ontario, Canada L5K 1B2

ABSTRACT

As CANDU nuclear power plants become more complex, and are operated under tighter constraints for longer periods between outages, plant operations staff will have to absorb more information to correctly and rapidly respond to upsets. A development program is underway at Atomic Energy of Canada Limited to use expert systems and interactive media tools to assist operations staff of existing and future CANDU plants. The complete system for plant information access and display, on-line advice and diagnosis, and interactive operating procedures is called the Operator Companion. A prototype, consisting of operator consoles, expert systems and simulation modules in a distributed architecture, is currently being developed to demonstrate the concepts of the Operator Companion. Specialized advisors are also being developed using expert system technology to meet specific operational and design needs.

INTRODUCTION

Atomic Energy of Canada Limited (AECL) has been a pioneer in the application of digital computer control to CANDU (<u>CAN</u>ada <u>D</u>euterium <u>U</u>ranium) nuclear power plants [1-3]. The next generation of CANDU plants will extend this leadership in the use of computer technology to the areas of:

- distributed control systems and data highways,
- computerized safety systems,
- advanced control room design,
- decision support systems, and
- advanced signal processing.

Recent developments in computer technology offer significant new opportunities to enhance nuclear plant safety and to protect the investment by assisting and counselling the plant operations staff in making routine and abnormal event decisions. Benefits to owners and operations staff will be achieved through:

- reduced safety and licensing concerns by providing enhanced plant monitoring systems and decision aids,

- reduced operations-staff stress from fewer but more informative alarm messages,
- streamlined operational tasks, and
- increased reliability and reduced operational costs through automated testing and predictive maintenance.

Development work is underway within AECL to use expert systems and interactive media tools to assist operations staff in existing and future CANDU plants. The complete system for plant information access and display, on-line advice and diagnosis and interactive operating procedures is called the Operator Companion [4]. Key functions that the Operator Companion will address include alarm annunciation, fault detection and diagnosis, plant status and vital parameter monitoring, and 'smart' operating procedures. AECL is working closely with CANDU utilities in Canada to select the most urgent and beneficial Operator Companion applications and to incorporate operating experience into them.

Nuclear plant staff continually access plant information, such as plant data, procedures, flowsheets, operational constraints, equipment status and history, when performing tasks ranging from system check-outs to reactor shutdowns. Discussions with the operations staff of CANDUs have led to the recognition that the amount of information required to operate a reactor continues to increase. In response to this potential problem, station personnel have indicated a need for enhanced support to operate the reactors to their full potential.

This paper provides an overview of the work currently underway and being planned at AECL on developing a prototype Operator Companion, and on developing application-specific advisors using expert system technology.

OPERATOR COMPANION OVERVIEW

The Operator Companion is conceived as a family of expert systems and other advanced computing systems communicating with each other and with the plant via a local area network (see Figure 1). This architecture offers a number of advantages:

- a distributed computing network provides the necessary multiprocessor, multitasking environment required to implement various strategies for multiple subsystems,
- the data from the system can be made available in preprocessed form to match the diagnostic strategy or strategies being considered,
- some modules can be dedicated, faster than real-time processors for plant data analysis,
- real-time simulators or plant analyzers can be incorporated for on-line power plant decision making and 'on-line' operations staff training,
- redundant workstations can be used for recovery from equipment failure,
- a modular approach provides greater resistance to obsolescence as technology evolves, and
- a modular development and implementation strategy can be used.

The following technical activities for the Operator Companion project evolved from discussions with CANDU design and operations personnel.

Improved CANDU Alarm Annunciation Strategy

An improved alarm processing and annunciation strategy for CANDU plants is required to assist the control room operator in obtaining a better and faster understanding of plant upsets. Some remedies, such as alarm conditioning, sequence-of-event recorders, classification of major and minor alarms, and sorting alarms by system, have already been implemented into the design. Additional improvements are still required.

On-Line Fault Detection and Diagnosis

This activity addresses the broad problem of on-line fault detection and diagnosis for any event, and providing advice to the operations staff on the most effective course of action. Fault detection has traditionally been handled by interpretation of alarms. However, diagnosis of the root problem is done manually by the operations staff performing a time-consuming search through flowsheets and manuals. The computer's ability to continuously and exhaustively search through all plant data offers the prospect for more automated fault detection and diagnosis and for providing a predictive capability.

Plant Configuration and Equipment Status Monitoring

Advanced computer and interface tools will be used to enhance the ability of the plant operations staff to monitor the physical status of the plant and major equipment. Plant personnel currently have to interpret the status of the plant from diverse information sources such as operations reports, manuals, drawings, control panel displays, and alarm indicators. On-line access to this information using advanced display techniques will provide a better indication of the plant status on which to base decisions.

Vital Operating Parameters

Expert system tools and novel information presentation methods will be used to provide the operations staff with a concise picture of the overall plant profile based on key operating parameters, normal and safety limits and other essential data.

Plant Operating Procedures

The aim of this work is to provide the plant operations staff with convenient and rapid access to relevant operating procedures and plant data as events unfold, and could include the use of adaptive procedures based on the state of the plant.

Specialized Operational Tasks

Expert system technology offers the opportunity of capturing scarce expertise and making it readily available to operations staff. Specific examples include fuel management, interpretation of plant chemistry data and identification of fuel defects.

REQUIREMENTS FOR THE OPERATOR COMPANION

Plant operations staff have identified the monitoring of plant configuration and equipment status, and the on-line detection and diagnosis of system faults as applications that can yield most immediate benefits. Functional requirements for each of these two areas have been developed and are summarized below. These requirements have been used to develop the prototype Operator Companion described later.

Several specialized operational and design tasks have been identified for immediate development to meet the needs of CANDU utilities. These systems are also described.

The remaining components of the Operator Companion will be developed in the next phase of the project.

Plant Configuration and Equipment Status Monitor

The configuration and equipment status monitor will provide a reliable and easily accessible record of field component status and plant operating profile. The system should offer plant personnel access to:

- a display of the flowsheets for the various operating areas of the plant,
- a display of the current status of all operable devices on the flowsheets,
- the ability to provide hard-copy prints of the flowsheets and operable devices,
- a display of the current device status, including physical state (i.e., Open, Closed, Throttled) and Work Protection Code tag state (i.e., red, yellow, green, or orange tags),
- the ability to perform simple modifications to the flowsheet displays to show temporary system changes, and
- the ability to display and modify a power supply list and/or an air supply list for applicable equipment.

On-Line Fault Detection and Diagnosis

The on-line fault detection and diagnosis system should:

- act as an "intelligent" operator assistant,
- monitor plant components during normal operation to alert the operations staff of incorrect operation and to detect changes in data profiles that indicate incipient component failure or deterioration of economic performance, and
- reduce information overload during abnormal plant transients, to diagnose their causes, and to suggest the appropriate human response to correct the situation.

IMPLEMENTATION OF A PROTOTYPE OPERATOR COMPANION

Overview and Application

The prototype Operator Companion has been implemented using distributed workstations linked on a local area network (LAN). Three modules have been developed and are connected by the LAN: a plant database, an operator console and a subsystem advisor (see Figure 2).

The requirements for the plant configuration and equipment status monitor lead to the need for a plant database and an operator console. The plant database acts as a central data repository for :

- raw data, such as sensor readings, in a form that is usable by other Operator Companion modules,
- component status (open, closed, work protection code, etc.) and trend data,
- component maintenance records,
- component specifications, and
- shift logs.

The operator console serves as a high-level interface to automatically monitor selected data from the plant database, and as an intelligent interface to the subsystem advisor module described below.

The requirements for on-line fault detection and diagnosis lead to the need for subsystem advisors dedicated to specific subsystems of the plant. These advisors, which are based on advanced

computer technology, such as expert systems, communicate with the operations staff through a message facility linked to the operator consoles or through more conventional alarm displays.

The network can contain any number of operator consoles and advisor stations. Each workstation can be dedicated either to monitoring specific parts of the plant or to meeting the needs of a specific operations group (e.g., operators, maintainers, etc.). One workstation of each type has been implemented for the prototype.

The Slowpoke Energy System (SES) heating-type nuclear reactor [5] has been selected as the trial application to develop the prototype Operator Companion as:

- the system contains a sufficient number of components, sensors and alarms to adequately test the concept, yet is not overly complex (so as to minimize the development effort required to cover the system), and
- the control computer provides access to the plant data.

Figure 2 shows this reactor with its own dedicated data acquisition system (DAS) together with the structure of the prototype Operator Companion. The reactor and its data acquisition system are considered to be independent of the Operator Companion; the only connection between the two systems being through a high-speed, uni-directional communication link. This link allows data collected by the DAS to be transferred periodically to the plant database for use by Operator Companion modules.

Networking is a key component of the Operator Companion and LAN's offer a means of connecting low-cost systems into a more powerful distributed architecture. The LAN approach is seen as offering the Operator Companion numerous benefits:

- the cost of personal computers (PCs) is usually less than the cost of a large centralized system,
- the processing power of the newer PC-based workstations rivals that of minicomputer-based processors,
- a simple Operator Companion network can be expanded to include *multiple* operator consoles, *multiple* expert system advisors, *multiple* plant databases and fault-tolerant redundant networks, as the need requires,
- equipment failure (e.g., of a single operator console) would not seriously impair Operator Companion operation. Redundant workstations and networks can offer failure-recovery possibilities, and
- a distributed approach provides greater resistance to obsolescence as technology evolves.

Development of the prototype Operator Companion has been centered around the Digital Equipment Corporation (DEC) VAXstation computer for the plant database, the Apple Macintosh family of personal computers for the network workstations, and Ethernet for the local area network. This configuration is in the final stages of integration and will be undergoing evaluation shortly.

Plant Database and Operator Console

The plant database has been implemented using the ORACLE relational database, while the operator console has been developed using HyperCard on the Macintosh personal computer. Specific functions built into the prototype console to meet the requirements of a plant configuration and equipment status monitor role, include:

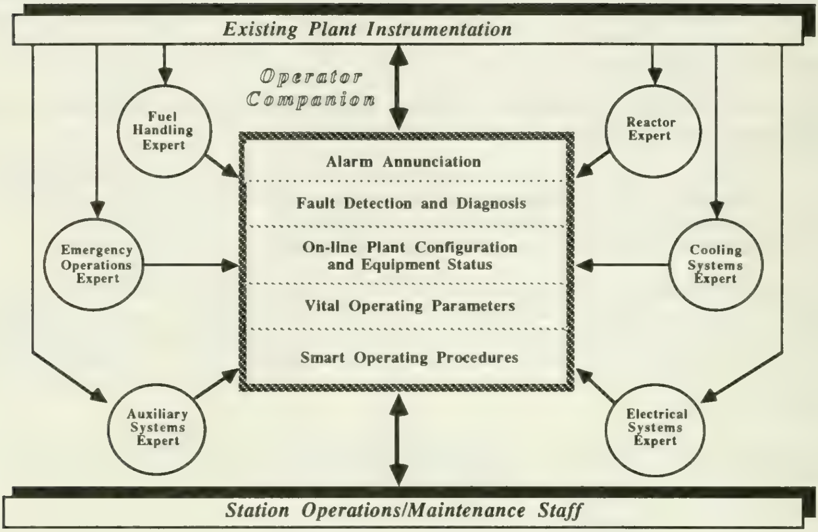


Figure 1. System architecture for the AECL Operator Companion.

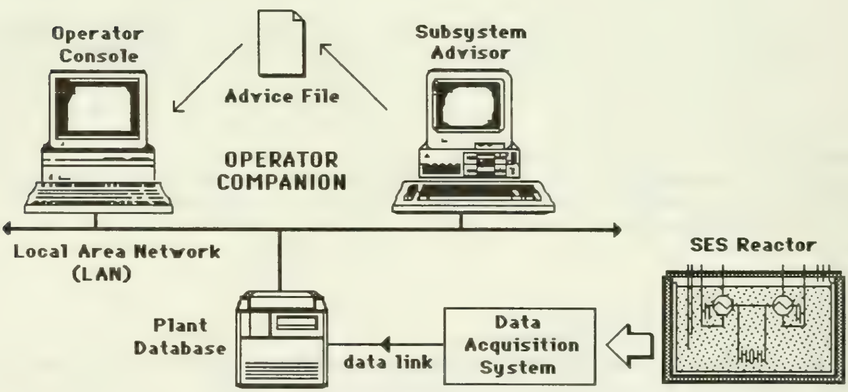


Figure 2. Overview of the prototype Operator Companion and the demonstration application.

- an overall process system overview, including on-line help and search-and-find capability (see Figure 3),
- an intelligent subsystem overview, including a live representation of component status information read from the database (see Figure 4),
- specific component data such as log data, specifications, trends and pictures, available on demand,
- the ability to change the physical state (open, closed, etc.) and work protection tag state (red, yellow, green, etc.) of a component in the plant database (though not directly controlling the component in the field),
- the ability to print any screen of information to a high-quality printer, and
- the recording of all component logs and network users into a plant shift log.

The interface to the subsystem advisor has been achieved through the use of advice records within the plant database. If a change is detected by the advisor, the operator is requested to check the advice record for further information.

Subsystem Advisor

The expert system shell NEXPERT OBJECT has been used for the subsystem advisor. A diagnostic knowledge base has been developed that monitors system parameters in the central plant database for one of the four subsystems in the SES reactor. Diagnostic messages generated by the advisor are dropped into a 'mail box' in the central database where their presence is detected and indicated to the operations staff on the operator console.

A symptomatic or shallow rule base of approximately 70 rules has been implemented for the Secondary Heat Transport System (SHTS). At present, the expert system acts much like a shadow for the control system; however, message generation has been enhanced to provide more information.

SPECIALIZED OPERATIONAL AND DESIGN ADVISORS

Several specialized operational and design advisors are being developed using expert system technology to meet the requirements of CANDU owners. Many of these advisors are being developed jointly with the owner utility. Each advisor is described briefly below.

On-Power Refuelling Application (FUELEM)

CANDU reactors are refuelled while operating at full power. FUELEM preselects channels that are candidates for refuelling based on channel power, fuel burnup distribution, refuelling rates, regional power distribution and recent refuelling history (based on the output from a fuel management computer code). The system has been demonstrated and is currently being customized for use at the Point Lepreau NGS.

Fuel Defect Detective

Defect Detective is an expert system to automate and improve the evaluation and location of fuel defects. The system quickly gives an assessment of the seriousness of the defect and an identification of its location so that the defective fuel can be removed as quickly as possible. The program can be easily modified to incorporate different evaluation and location criteria. The system is in use and is undergoing further development.

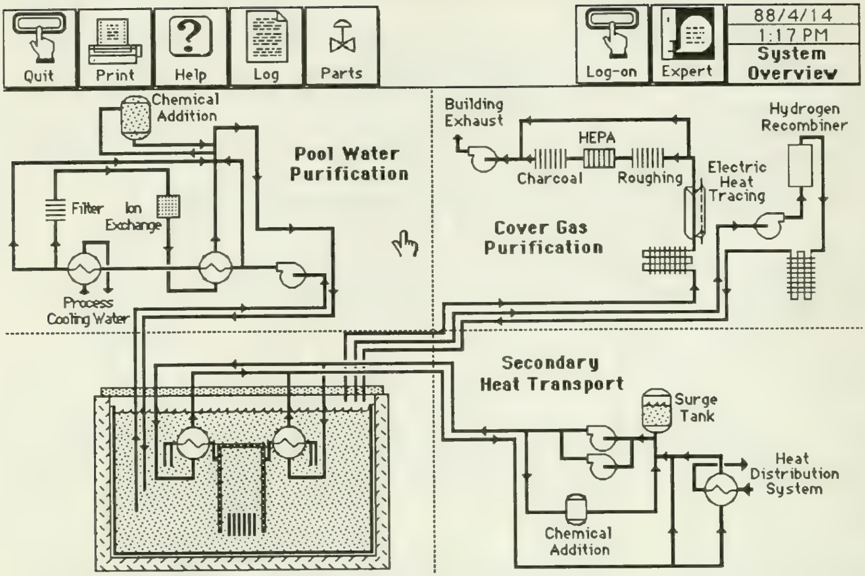


Figure 3. Process system overview of the configuration and equipment status monitor console.

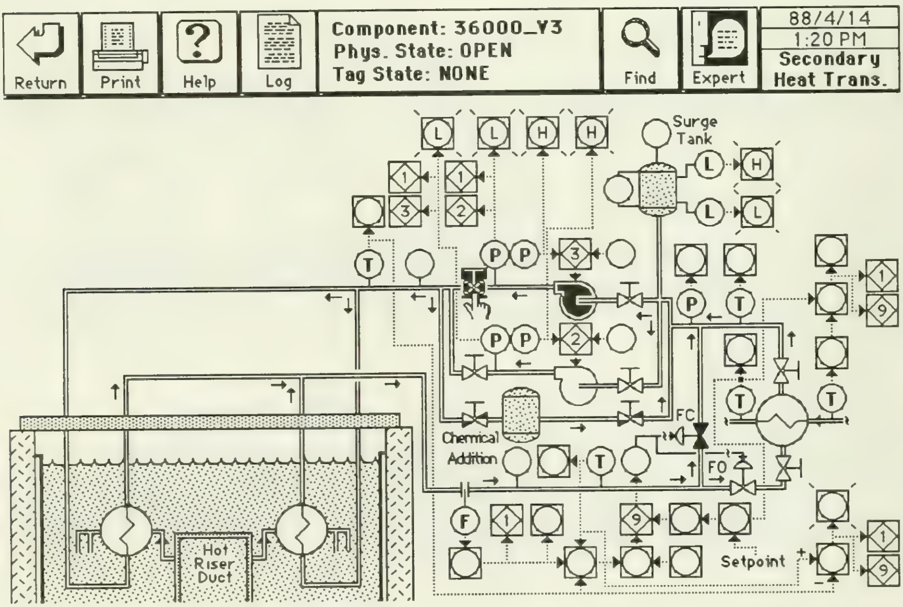


Figure 4. Process subsystem overview of the configuration and equipment status monitor console.

Fault Diagnosis for Programmable Digital Comparators

Programmable digital comparators (PDC's) are used in the shut down systems of some CANDU reactors and are generally very reliable. Faults in these comparators must be diagnosed and repaired in a timely manner since they impair the shut down system. The expertise for diagnosing faults resides with a limited number of staff at a station and there is a concern that this expertise will be diluted or lost with staff turnover. An expert system is being developed to capture this knowledge and to make it more widely available within a given station. It is expected that this improved capability for diagnosis will reduce unnecessary board swapping that takes place during maintenance, thereby leading to less wear on components within the PDC's.

Condenser Sea Water Leak Advisor

Plants that are cooled by sea water require the use of procedures to minimize the effects of corrosion to the condenser, feed train, and steam generators in the event of a sea water leak into the condenser. An expert system, with access to station chemistry data, is being implemented to advise on the required action (e.g., derating, shut down or cool down) in the event of such a leak.

Shut Down System Diagnostic Advisor (SADAU)

The operations staff at Hydro Quebec's Gentilly 2 NGS have identified the need for a post-trip diagnostic support system for the plant's shut down systems. SADAU will help the Shift Supervisor to evaluate the root cause of the plant upset condition following a trip. System knowledge will be based on information contained in the shut down system and related process system design manuals. This knowledge will also incorporate the experience of operations staff, information from significant event report (SER's) and other system design information. SADAU will be integrated into the Operator Companion for future CANDU plants.

Plant Emergency Operating Procedures (EOP's)

Development work is underway to use expert system technology and interactive media (e.g., hypermedia) tools for a computerized EOP system. The aim of this work is to provide the operations staff with rapid access to emergency operating procedures, plant data, and context-sensitive support information as events unfold. The use of adaptive procedures, based on the state of the plant, is planned. A prototype has been developed using interactive media tools. The system will be integrated into the Operator Companion.

Shell-and-Tube Heat Exchanger Design Advisor

A prototype advisor has been developed to investigate the feasibility of using expert systems to aid junior process system designers with the selection of components for shell-and-tube heat exchangers. The selection criteria for heat exchanger design are based on process, environmental and administrative constraints. The system consists of approximately 140 rules.

CONCLUSIONS

The Operator Companion project has allowed us to experiment with advanced computer technology that offers significant opportunities to mitigate operational problems in nuclear power plants. The Operator Companion concept, consisting of a networked arrangement of engineering workstations communicating with each other via a plant database, has been demonstrated. Expert system technology and interactive media tools are the key technologies used for assisting operations staff. A functioning plant configuration and equipment status monitor console, and a subsystem advisor have been developed to prove the concept. Work continues on the development of a full-scale Operator Companion.

Specialized advisors, based on expert system technology, are also being developed to meet specific operational and design needs. These advisors will be integrated into the design of the next generation of CANDU reactors to improve their operating reliability and safety.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the contributions of J.J. Lipsett, A. Natalizio, R.A. Olmstead, R.R. Shah, H.E. Sills and J.E.S. Stevens towards the development of the concept for the Operator Companion. The contributions of the operations and technical staff at Hydro Quebec's Gentilly 2 NGS and at New Brunswick Electric Power Commission's Point Lepreau NGS are also acknowledged.

REFERENCES

1. D. Chan, J. Pauksens and J.R. Popovic. "Large-Scale Implementation of Computerized Plant Control and Monitoring in CANDU Plants." Proceedings from the Seminar on Integrated Power Plant Computer Communications, San Francisco, CA., August 1986.

2. J. Pauksens, J.R. Popovic and S. Groom. "Development of Computerized Man-Machine Interface Systems for CANDU Nuclear Power Plants." Proceedings of the ISA Symposium on New Technologies in Nuclear Power Plants, Instrumentation and Control, Washington, D.C., November 1984, pp. 373-384.

3. E.M. Hinchley, N. Yanofsky, J.D. Beattie and E.F. Fenton. "The CANDU Man-Machine Interface and Simulator Training." Atomic Energy of Canada Limited Report, AECL-7768, 1982.

4. A. Natalizio, J.W.D. Anderson and H.E. Sills. "Operator Companion: Advanced Support Systems for Plant Operators." Atomic Energy of Canada Limited Report, AECL-9612, 1988.

5. J.W. Hilborn and G.F. Lynch. "Slowpoke - Heating Reactors in the Urban Environment." Atomic Energy of Canada Limited Report, AECL-9736, 1988.

METHODOLOGIES

A V&V Program for a Real Time Operator Advisor Expert System

B. K. HAJEK, C. R. HARDY, D. W. MILLER, and R. BHATNAGAR
The Ohio State University
Nuclear Engineering Program
Robinson Laboratory
206 West Eighteenth Avenue
Columbus, Ohio 43210, USA

ABSTRACT

An Operator Advisor (OA) consisting of four integrated expert systems has been under development at The Ohio State University since 1985. The OA was initially conceived as two independent systems, one to manage procedures for the operator, and the other to diagnose plant faults and to validate sensor data. The OA has used the Perry Nuclear Power Plant simulator as the reference plant. It has been programmed with knowledge gathered from power plant experts and plant documents. Following incremental programming, the OA has been tested off line by running scenarios on the plant simulator and verifying that the OA responded as predicted by the simulator. Specifications have been developed and periodically published in the literature over the past four years. These are currently being collected and expanded upon as part of a formal Verification and Validation (V&V) Program. The V&V Program is defined by three types of documents: specification documents, a V&V document, and procedures to support implementation of the specifications and V&V program.

INTRODUCTION AND BACKGROUND

An Operator Advisor has been under development at The Ohio State University for the past four years. The Operator Advisor is a combination of four expert systems. The first to be developed was a Dynamic Procedure Management System (DPMS) (1, 2). The second was a Diagnostic and Sensor Validation System (DVS) (3). The next step in development was to combine these two expert systems into a single system. Prior to doing this, however, it was necessary to determine what actions power plant operators performed to assure that a system would be developed that would be responsive to their needs.

Generalized Task Analysis

A generalized task analysis of the actions performed by reactor operators during nuclear power plant operations was conducted. Specifically, their actions in response to abnormal plant conditions were analyzed (4, 5). This analysis resulted in the identification of several high level tasks, which are to:

1. Monitor and comprehend the state of the plant.

   This task is usually performed in the background. That is, operators are usually busy with several control room duties such as working with maintenance personnel. Actual monitoring is done by the plant instrumentation and alarm systems. During power changes or evolutions involving the starting or changeover of equipment, operators must directly observe parameters and identify whether the evolution is progressing normally.

2. Identify normal and abnormal plant conditions.

   During normal steady state operations, reactor operators are able to perform several responsibilities. When an alarm sounds, operators change tasks, identify the abnormal condition causing the alarm, and respond to the alarmed condition. During normal transient evolutions, operators will identify normal and abnormal conditions by direct observation as well as by alarm response.

3. Identify abnormal conditions.

   When an abnormal condition is identified, operators determine the underlying reason for the condition in order to appropriately respond as effectively as possible. In some cases, they must respond independent of the cause of the abnormality in order to maintain the safety of the plant.

4. Predict plant response to specific control actions to be taken in response to diagnosed abnormal conditions.

   Prior to taking an action, operators anticipate the expected result. That is, for example, if a pump has tripped, they have an expectation that starting a standby pump will restore pressure to the system discharge header. They also anticipate the next action that will be required should the standby pump fail to start.

5. Then select and implement the best available control action to mitigate the abnormal condition, and monitor the results of that action, and

6. Determine backup control actions in the event of a failure of the primary action.

Thus, an effort was begun to develop the Operator Advisor as a single system that would perform the monitoring and diagnosis functions, and that would advise (provide) the operator with a procedure to follow for fault mitigation, monitor the performance of the procedure, and advise the operator on backup steps to follow should the primary procedure fail.

<u>System Architecture</u>

This effort has resulted in a system to monitor all available nuclear plant process variables and alarm states, diagnose deviations from normal operations, and advise the plant operators

on action to be taken to correct the abnormalities. In the process, two additional expert systems have been developed.

A Plant Status Monitoring System (PSMS) was developed to monitor plant variables and alarm states. This system also has the capability to perform simple diagnosis, and to direct the activities of the two previously developed expert systems (6, 7).

Finally, an interface with the plant data was developed. This interface is an intelligent database system that receives and manages data supplied by the plant computers (8). The goal of this development effort was to enable the Operator Advisor to make use of as much pre-processed data as possible. Further abstraction is performed as needed in the intelligent database.

The architecture of the Operator Advisor is shown in Figure 1.

The current effort involves expanding the Operator Advisor to perform diagnostics and fault management on a broader scope of plant systems, implementation of a formal Verification and Validation (V&V) program, and interfacing the Operator Advisor to receive input directly from a plant referenced simulator.

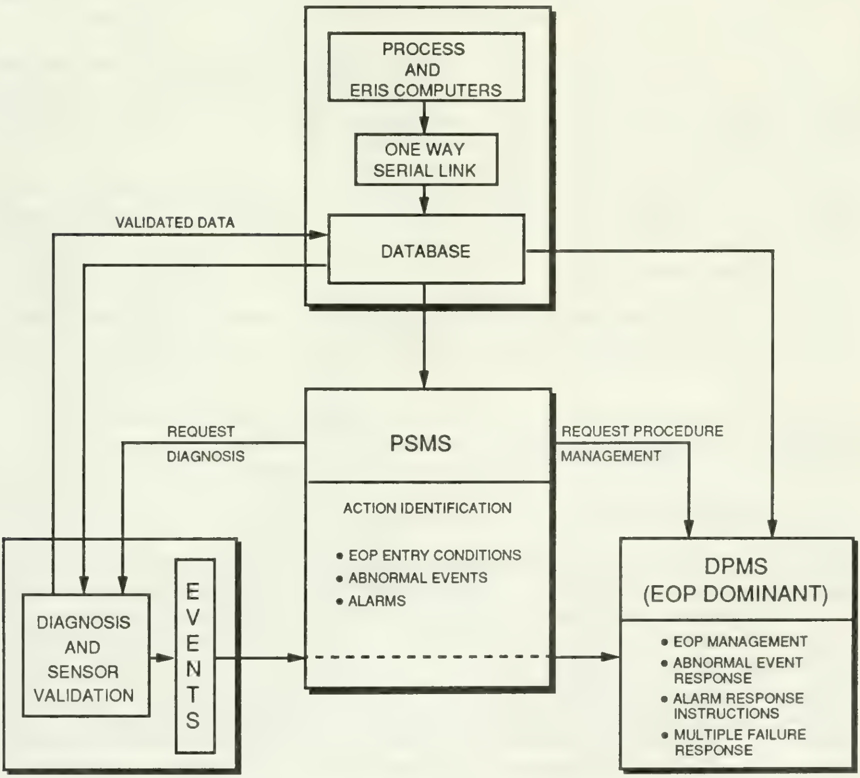


Figure 1. Architecture of the Operator Advisor

Reference Plant and Plant Interface

The Operator Advisor consists of the system shell and the knowledge. While the shell is designed using the Generic Task Approach developed at The Ohio State University Laboratory for Artificial Intelligence Research (LAIR) (9), the knowledge must reflect the procedures and components of a real plant for the OA to be testable.

For this reason, a plant with an operational plant referenced simulator was sought. The Perry Nuclear Power Plant was selected because it is close to Ohio State, is heavily instrumented, has a state of the art Safety Parameter Display System (the General Electric Emergency Response Information System (ERIS)) for pre-processing of data, and because key members of the development team are familiar with the BWR system.

The interface with the plant will be through the two plant computer systems. One of these is the plant Process Computer, and the other is ERIS. The Process Computer provides raw plant data for each sensor that is monitored. ERIS provides pre-processed data for a number of variables that are necessary for determining the state of the plant for safety level monitoring. For example, rather than providing eight reactor level values, ERIS provides a single average and validated value.

However, prior to interfacing the Operator Advisor to the plant, an interface to the plant referenced simulator will occur. This interface will be to the simulator computers.

Installation of the Operator Advisor on the Perry simulator involves the interfacing of a Sun 4 expert system work station with the two Gould 32/55 minicomputers that perform the simulation process, and the Gould 32/77 minicomputer that is used to perform the ERIS simulation.

The Gould 32/77 also receives data from the simulator computers through shared memory. Approximately 250 analog inputs, 250 analog outputs, and 1000 digital signals from the plant simulator are stored in a common area of memory. All simulator data is accessible through this common area.

Serial communications software is currently being developed to control the flow of data, to assure the maintenance of data integrity, and to transfer binary data packets in a logical sequence.

The data will be transferred across a one-way link to assure that the expert system will not interfere with the operational integrity of the simulator.

V&V OF EXPERT SYSTEMS

V&V of expert systems is different from that of conventional software because both the knowledge base and the coding must be considered. Since expert systems are normally built with a prototyping (or rapid prototyping) process, as has our Operator Advisor, the V&V effort should be started early in the development process. Then, as the knowledge base expands, V&V must be repeated continually.

Before discussing V&V further, it is necessary to define specifically what is meant by the two terms - verification and validation. For this, we turn to two recent EPRI publications (10, 11) that describe the V&V process we are attempting to implement.

> *Verification pertains to the internal correctness of the system and addresses the task of eliminating errors made by the builders in translating their original plan into a detailed design and coding it. Validation, on the other hand, refers to external correctness which is manifest in correct or desired output when the system is operating in a realistic environment.* (10)

This was stated more descriptively in (11) where it is said that

> *. . . verification assures the system is working properly as designed, while validation assures that the system actually helps the user as intended in the requirements and specifications.*

Both representations of V&V allude to the need for system specifications to be developed prior to coding.

A formal V&V effort should progress from the development of system specifications to coding, then to verifying that the coding meets the specifications. An iterative process should occur with additional coding being performed followed by additional verification after each incremental coding step. Validity of the system should be embodied in the specifications, and also should be tested periodically during the coding to assure that the system continues to approach the needs of the intended users.

SPECIFICATIONS FOR THE OPERATOR ADVISOR

Development of the Operator Advisor began with a limited set of specifications for a procedure management system based primarily on the concept of maintaining the critical safety functions (12) of the plant. It was also decided to integrate the recently (at that time) introduced symptomatic emergency operating procedures into the system. Following initial prototyping, these specifications were published (1, 2) in the normal course of typical university research.
Shortly after the procedure management effort began, a second effort was started to build an expert system diagnostic aid. Again, safety function maintenance was established as the primary specification, and the plant emergency, abnormal, and operating procedure structure was to be considered when establishing the malfunction hierarchy. But no other formal specifications were established. The design of this system was also published in the open literature (3, 4).

It was also about this time that we conceived a need and a process to integrate the diagnostic and procedure management systems into a single system that would eventually become the Operator Advisor. A generalized task analysis was performed, and a structure was proposed for the development of an expert system that would automate the operator tasks of monitoring the plant status and diagnosing faults, and that would then advise the operator on a procedure to follow, monitor his progress through the procedure, and advise on the implementation of backup steps should a primary step fail. The task listing and structure were also published (4, 6, 7).

Until recently, this set of papers has served as the primary specifications for continuing development of the Operator Advisor. They have now been compiled into a paper that details the purpose of the system, the structure, and the software of the Operator Advisor (13). But this paper was not meant to, and doesn't, fulfill the need for a formal set of specifications.

Formal specifications are currently being developed. They follow the general guidelines provided in EPRI NP-5978. Three specification documents are being prepared.

The first document includes the general specifications for the application of the Operator Advisor to assure that it meets user needs. This document will provide the basis for the validation of the Operator Advisor.

The second document includes the detailed system specifications. It specifies how each of the general specifications should be met at the functional level. This includes the division of the Operator Advisor into its sub systems, specifications for the operation of each sub system, specifications for interconnections and communications, hardware specifications, man-machine interface specifications, and data/knowledge input specifications. This document will be used by the system programmers to translate the system requirements into code, and by the V&V team as the test standard.

The third document will include the software design specifications, including requirements for specific tools to use, and rule and object structures. It will describe how the Operator Advisor shall be built. It will specify requirements for programming procedures, and will serve as the standard for the exhaustive testing of the rules, objects, inference strategies, and communications paths.

V&V OF THE OPERATOR ADVISOR

We have performed the V&V function during our earlier development work by comparing the operation of the Operator Advisor to the sequence of events that occur on the Perry simulator following initiation of an hypothesized malfunction. However, this process has not yet been effectively formalized, it has not included sufficient involvement from the plant operators, and it has not been done with the Operator Advisor receiving continually updated data from the simulator (that is, the Operator Advisor has not been on line at the simulator).

In this context, we have concentrated on the Verification side of the V&V equation. We have verified that the results of the expert system analysis are correct relative to the input parameters, and the emphasis in these efforts has been on DVS. We have not yet determined whether the results will be available in a format or in a time frame to be of use to the operators.

Preparation of the V&V Plan

In our current efforts, we are moving to correct these deficiencies, and to follow the general guidelines for V&V of expert systems recently published in EPRI NP-5978. The specifications for the Operator Advisor are being developed with their testability being kept in mind.

A document is being prepared to detail the V&V Plan. This document will describe the V&V process and the V&V team membership; the procedures to follow to confirm that specifications can be tested, audited, and are being met; software tests; knowledge base tests; and documentation requirements.

V&V Activities

To effectively verify and validate the Operator Advisor, it will be necessary to expand the number of systems and malfunctions monitored by the Operator Advisor, and to increase the involvement of the plant operators in the testing process.

The Operator Advisor development to date has concentrated on the plant feedwater system to validate the feasibility of the approach. For implementation of the system at the plant site, the knowledge base will be expanded to include additional systems and additional malfunctions. This task, which expands the utility of the Operator Advisor, is proving to be an effective means of bringing new team members on board. It introduces them to each aspect of the expert system, and enables them to rapidly gain expertise in the operation of a small portion of the nuclear plant as they learn how to succinctly organize their thought processes.

Development of this type system involves the hypothesis of malfunctions, or the hypothesis of malfunction scenarios. As each scenario is outlined and entered into the knowledge base, it is tested on the simulator to verify and validate the correctness of the knowledge base (database contents), and correct operation of the diagnosis and procedure management systems.

A future part of the process will include monitoring of human operator performance as tasks are performed with and without the Operator Advisor. Results of this analysis will be used to improve the Operator Advisor and the man-machine interface.

Only by testing a large number and variety of scenarios can we verify that the Operator Advisor works properly under all design conditions. Likewise, it is only through extensive running of scenarios that we can validate the Operator Advisor's ability to actually help the human operator perform his duties.

Other tests will be performed to check the programming by assuring that each and every node in the Operator Advisor properly fires when the appropriate conditions are simulated, and that the rules and objects all meet the requirements and software specifications. These are the only possible exhaustive tests.

Regulatory Approval

Another concern we have about implementing an effective V&V process is in eventually achieving regulatory approval. This effort actually began when we first started developing the individual modules for the Operator Advisor when the initial objectives for the system were specified. Verification that the system operates according to these specifications has occurred as off line tests have been run using the Perry simulator.

As we enter the implementation phase, the V&V process requires further formalization as previously stated, and the validation phase must be initiated. Since the Operator Advisor is expected to provide high level conclusions and recommendations to the human operators, it is expected that the V&V process will need to follow the more rigorous NRC guidelines normally reserved for systems such as the reactor protection system. Success in obtaining regulatory approval will depend on the quality and integrity of the V&V process. To maximize the probability of success, we will be rigorously following the life cycle V&V process detailed in EPRI NP-5978.

We anticipate that most V&V efforts can be completed on the plant simulator, including integrity checks, integration tests, acceptance tests, and field modification tests. However, we recognize that plant specific simulators do not model all plant components and control actions, that those modeled are not always of perfect fidelity, and that additional modifications will be necessary for installation on the actual plant. Thus, in anticipation of the regulatory approval process, a plan will be formulated to modify the code and to V&V these modifications for the installation phase.

PROCEDURES

The specifications and the V&V Plan state the requirements for design of the Operator Advisor. Procedures need to be developed to precisely and accurately implement these requirements. We have informally used many unwritten procedures to date. This has been possible because of the limited size and closeness of our development team. However, as development is continuing and turnover (known as graduation in our environment) occurs, it is becoming more important to formalize the procedures for this process.

Each procedure will be referenced by the V&V Plan. This will likely be in an appendix to be used to audit the adequacy of procedures to assure the integrity of the development process.

Procedures are expected to fall into three general categories: (1) Those required to implement the specifications, (2) Those required for the user to interact with the Operator Advisor, and (3) Those required to implement and audit the V&V Plan.

The procedures will eventually be translated into additional documentation such as Operations and Maintenance Manuals for the Operator Advisor.

Specific procedures currently being considered or developed deal with source code commenting, rule and object structuring, software change and documentation requirements, user updating of the software and knowledge bases, verification tests, and data/knowledge acquisition and input. Only the last of these will be discussed in this paper.

Data/Knowledge Acquisition and Input

A key concern of the development team is the assurance that data is properly collected and properly coded into the database and the knowledge base. Thus, a key procedure will describe methods to obtain information from experts, methods to extract information from written material, such as plant procedures and operations manuals, methods for assuring the consistency of the data and knowledge, and methods for assuring the consistency of the knowledge representation in coding.

This element of the development process breaks down into two categories: (1) Data/knowledge acquisition, and (2) Data/knowledge encoding.

Data/Knowledge Acquisition.

Procedures are needed for the following aspects of data and knowledge acquisition:

1. Identification of experts/key expert

2. Resolution of conflicts among experts

3. Verification of most recent revisions

4. Requirements for documentation of source, revision number, date

Data/Knowledge Encoding.

Two aspects of encoding must be considered.

The first, and the one for which policy and procedures need to be established is that of the perspective and philosophy of the writers of the code. This should be stated in the General Specification, and then detailed further in the System Specifications. However, if it is not also emphasized in the procedures on how to enter information, the approach intended by the formulators of the expert system can quickly be lost or modified. The worst case scenario is that the knowledge will be entered with a mix of approaches for analysis, and it will be extremely costly to make the many changes that are likely to be needed.

As an example, we have specified that the Operator Advisor will detect deviations from normality. Thus, we are interested in detecting that water level is not within a normal range. We are not interested in detecting that water level is low. This is a very subtle difference, but it can have major impacts on the reasoning associated with more complex conditions where many (rather than just two possibilities such as high or low) possible abnormal states could occur. In particular, we might not identify an obscure state.

The second aspect of encoding involves the structure of the objects and rules themselves. They must conform to clearly specified styles and formats. This is easier to manage because they can be given structure, and this structure can be tested. Further, much of the testing may be automated with software that exhaustively tests each rule and object.

SUMMARY

The V&V Program for the Operator Advisor consists of system specifications, a V&V Plan, and procedures for implementing the specifications and the V&V Plan.

Ideally, a V&V Plan would have been put into place prior to any coding of the Operator Advisor. This would have included development of the general and functional specifications for the system, as well as the procedures for implementing them.

However, because of the research and development environment in which the Operator Advisor has been developed, this process was not followed. We are now formalizing the process that has been in place during the past several years. It is anticipated that this formalization will assure a precise and accurate implementation of the general specifications for the Operator Advisor, and will assure that the completed system will be an effective aid for control room operators.

Another important aspect of implementing an effective and well documented V&V Plan is to assure success in the regulatory approval process.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the National Science Foundation (Grant No. CBT-8400840) and the U.S. Department of Energy (Grants No. DE-AC02-86NE37965 and DE-FG07-88ER12819) for their support in the development of the Operator Advisor.

We also would like to express our appreciation to the staff of the Perry Nuclear Power Plant Nuclear Training Department for their extensive assistance in obtaining plant data and for working with us to run plant transients on the Perry simulator. This collaboration has provided a reference plant for the work we are performing.

The collaboration of GE Nuclear Energy, Electronic and Computer Services, also is appreciated. GE's collaboration is making it possible to assure that the Operator Advisor will properly interface with the plant computer systems.

Finally, the authors wish to acknowledge the support provided by The Ohio State University.

REFERENCES

1. Sharma, D. D., Miller, D. W., and Chandrasekaran, B., "Design of an Artificial Intelligence System for Safety Function Maintenance," Transactions of the American Nuclear Society, Vol. 50, pp. 294 - 297, 1985.

2. D.D. Sharma, D.W. Miller, B.K. Hajek, and B. Chandrasekaran, "Intelligent Process Control Operator Aid - An Artificial Intelligence Approach," in Proceedings of the Sixth Power Plant Dynamics, Control and Testing Symposium, April, 1986.

3. Hashemi, S., Hajek, B. K., Miller, D. W., Chandrasekaran, B., Punch III, W. F., "An Expert System for Sensor Data Validation and Malfunction Detection." presented and published in the Proceedings of the ANS Conference on Artificial Intelligence and other Innovative Computer Applications in the Nuclear Industry, Snowbird, Utah, 1987.

4. B.K. Hajek, D.D. Sharma, S. Hashemi, B. Chandrasekaran, and D.W. Miller, "AI Enhancements to Safety Parameter Display Systems," in Proceedings of the International Topical Meeting on Advances in Human Factors in Nuclear Power Systems, April, 1986.

5. D. W. Miller and B. K. Hajek, "The Knowledge-Based Framework for a Nuclear Power Plant Operator Advisor." Accepted for presentation at the Annual Meeting of the American Nuclear Society, Atlanta, GA, June, 1989.

6. Hajek, B. K., Stasenko, J. E., Hashemi, S., Bhatnagar, R., Punch III, W. F., and Yamada, N., "The Structure of an Expert System to Diagnose and Supply a Corrective Procedure for Nuclear Power Plant Malfunctions," presented and published in the Proceedings of the ANS Conference on Artificial Intelligence and other Innovative Computer Applications in the Nuclear Industry, Snowbird, Utah, 1987.

7. B.K. Hajek, D.W. Miller, R. Bhatnagar, J.E. Stasenko, W.F. Punch III, and N. Yamada, "A Generic Task Approach to a Real Time Nuclear Power Plant Fault Diagnosis and Advisory System," in Proceedings of IEEE International Workshop on Artificial Intelligence for Industrial Applications, Hitachi City, Japan, May, 1988.

8. Bhatnagar, R., Gandikota, M. S., Davis, J. F., Hajek, B. K., Miller, D. W., and Stasenko, J. E., "An Intelligent Database for Process Plant Expert Systems," presented and published in the Proceedings of the ISA 88 International Conference and Exhibit, Houston, TX, October, 1988.

9. B. Chandrasekaran, "Generic Tasks in Knowledge-Based Reasoning: High Level Building Blocks for Expert System Design," IEEE Expert, Fall, 1986, pp. 23-30.

10. Kirk, D. B., Murray, A. E., "Verification and Validation of Expert Systems for Nuclear Power Plant Applications," EPRI NP-5978, August, 1988.

11. Groundwater, E. H., Donnell, M. L., Archer, M. A., "Approaches to the Verification & Validation of Expert Systems for Nuclear Power Plants," EPRI NP-5236, July, 1987.

12. Corcoran, W. R., et al, "The Critical Safety Functions and Plant Operation," Nuclear Technology, Vol. 55, pp. 690-712, 1981.

13. Bhatnagar, R., D. W. Miller, B. K. Hajek, and J. E. Stasenko, "An Integrated Operator Advisor System for Plant Monitoring, Procedure Management and Diagnosis." Submitted for publication in Nuclear Technology, February, 1989.

A Knowledge-based Approach to Root-cause Failure Analysis

H. T. SU, L. W. CHEN, and M. MODARRES
Department of Chemical and Nuclear Engineering
University of Maryland
College Park, Maryland 20742, USA

R. N. HUNT* and M. A. DANNER
Baltimore Gas and Electric Company
Calvert Cliffs Nuclear Power Plant
Lusby, Maryland 20657, USA

ABSTRACT

This paper presents the results of a joint research effort being conducted by the Baltimore Gas and Electric Company and the University of Maryland. This research effort is devoted to extend the reasoning paradigm of an expert system, which uses a Goal Tree-Success Tree (GTST) knowledge-base model to analyze the root-causes of system failures. Because more comprehensive root-cause failure analysis often requires utilization of various experts, a Multi-GTST model is developed to meet this requirement. In the Multi-GTST model, a blackboard concept is applied to communicate information between GTSTs. Moreover, fuzzy logic theory is also incorporated to infer uncertain information.

Finally, the construction of the present expert system using top-down software design technique is described, and its current capability is discussed.

1. INTRODUCTION

Improvement in overall performance of engineering systems results from implementation of operational programs and design activities which ultimately improve the performance of individual components and systems. Historically, these programs relied very heavily on empirical evidence gathered during investigation of system failure. This approach, while easy, often leads to incomplete set of cause and effect, and cannot easily deal with the evolution of system operation as time goes by. A formalized model, however, does not exist to help the investigation and to put findings of such investigations (e.g., failure

*Currently with Idaho National Engineering Laboratory, Idaho Falls, Idaho 83415

mechanisms and environmental impacts) on perspective. Such a model will help to determine the performance of hardware, as well as root-causes of hardware failures. Additionally, effective automation of such a model on computers for applications in failure analysis procedures will be of great importance.

During the past six years the Baltimore Gas and Electric Company and the University of Maryland have been working together to develop a new tool which can be applied to resolve the above problem, especially in the assessment of plant reliability and safety. The center piece of this assessment has been the concept of Goal-Tree/Success-Tree (GTST) model which has been discussed in many of our previous publications [1,2]. In the past two years we have applied the GTST model for the analysis of root-causes of failures. Preliminary versions of our root-cause analysis work have been used for a variety of engineering applications [3,4]. In these versions, the GTST has formed the knowledge-base of an expert system which through some organized questions can hypothesize possible root-causes of failures. However, wider range applications of this expert system require utilization of various experts, each specializes in a narrow area of failures (e.g., failures occurred due to corrosion). In other words, a multi-knowledge-based expert system may be needed for a more comprehensive root-cause failure analysis problem. An expert system composed of Multi-GTST models is developed which meets this requirement.

In the Multi-GTST model, a common data area is used similar to a blackboard concept [5], through which all the information in different GTST's are communicated. If the outcoming solutions from a single GTST knowledge-base is not conclusive, then the expert system will generate "hypotheses". In order to prove these hypotheses, the expert system will select and search other appropriate GTST knowledge-bases which can support the "hypotheses". This process will be repeated until all the hypotheses are proven or the solution are satisfactory. Finally, the expert system will present a conclusive root-causes of failure. The Multi-GTST approach has enabled us to search for complex causes of failure by developing hierarchies of GTSTs. For example, a GTST for the major plant systems or components of concern, and separate generic GTSTs that model the basic mechanisms by which components fail (for example cracks, corrosion, or fatigue). When one or more of these mechanisms of failure is suspected, the respective GTST will be searched to establish the hypothesis. In this paper, we describe the implementation scheme of the new version of the Root-Cause Analysis Expert System as well as its application domain.

2. THEORETICAL BACKGROUND

In the development of this system, we have surveyed several models for representing a deep human cognitive knowledge, such as if-then-else rule, digraph [6] and Goal Tree-Success Tree. However, the Goal Tree-Success Tree (GTST) Model was found to be most appropriate to reflect process topology [3,4,7,8]. Furthermore, due to the uncertainty associated with evidences that lead to the establishment of a root-cause failure, we have incorporated the theory of fuzzy logic into the GTST model. These theories, GTST model and fuzzy logic, are described in the following.

GTST MODEL

Traditionally there have been difficulties in understanding and describing the properties of complex systems. Simon [9] has shown that complex systems evolve from stable simple systems and that the resulting systems will be hierarchical. Most physical systems are complex and exhibit hierarchical structure.

Goal Tree-Success Tree (GTST) model [3,4,10] which has a tree-like hierarchical structure has shown the capability to qualitatively provide the required ingredients for representing the deep-knowledge of a complex process system for problem-solving. Each node in the tree represents a goal, or more precisely a function in the case of physical systems. Upon developing the goal tree part of the GTST, the top plant goal or objective should be explicitly defined at first. The goal tree is then decomposed vertically downward from the objective in levels to progressively more detailed lower levels. At the bottom the hardware components (if ones) that achieve the lowest level goals are described. To assure the completeness of knowledge representation, two simple rules are applied.

1) When looking downward from any goal towards the bottom of the tree, it must be possible to define explicitly **how** the specific goal or subgoal is satisfied.
2) When looking upward from any subgoal towards the top objective, it must be possible to define explicitly **why** the specific goal or subgoal must be satisfied.

It is evident that a GTST is a success oriented model of how an overall objective is achieved. It is only a failure that ultimately causes the GTST goals to be lost from which we infer possible root-causes. A complete GTST is a graphical model showing how the individual goal and subgoals interact to achieve the overall objective or top goal. In the context of root-cause failure analysis the overall objective is "Prevention of failures". In what follows we describe how a GTST is used for reasoning about failures.

USE OF GTST FOR ROOT-CAUSE DETERMINATION

To find the root-cause of a failure, it is necessary to build a GTST for each system, component and protection against each mode of failures. This hierarchical model will ultimately have a structure typified by a graphical representation shown in Figure 1. The bases for this break down is the fact that the cause of a system failing to perform its function can be separated into the failure of the various components within the system. For example, failure of a pump results in the failure of the system to pump the working fluid. The cause of a failure for each component can likewise be separated into failures of individual piece parts associated with the component, e.g., failure of the impeller vanes results in the failure of the pump.

The failure of an individual piece part will be the result of its various possible failure modes, e.g., erosion of the impeller results in the failure of the impeller. Each failure mode can be modeled using the Goal Tree Analysis process. These models can be done in the general case to allow their use with many different piece parts having the same failure modes.

In order to reason with a GTST structure, each subgoal and goal is related to a single criterion or a set of criteria (e.g., evidences) from which the success or failure of the goal can be concluded. This information is supplied by the analyst. Simple questions are then

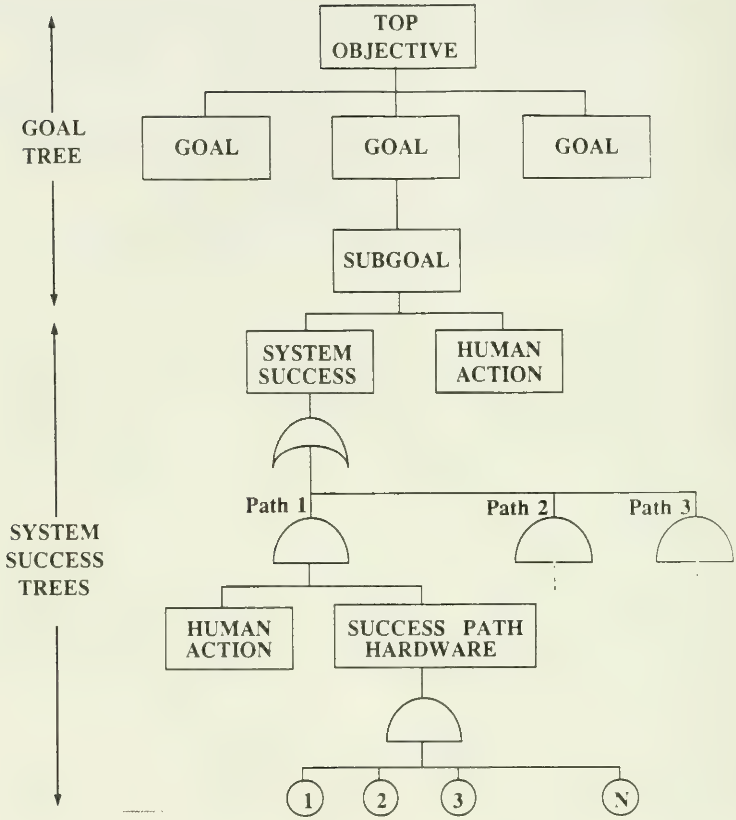


Figure 1. Typical Structure of a GTST Model

designed to relate the criteria required for the success of the goal or subgoal. The relationship of a "yes" or "no" answer to the success or failure of the process goal is described below:

A question may be (Y) or (N). A (Y) question means that a "yes" answer to it is interpreted as a success of the goal. Likewise, a "no" answer to a (N) question is also a success of the goal. A "no" answer to a (Y) question or a "yes" answer to a (N) question is interpreted as a failure of the goal. In cases that an answer is not known with high certainty, the answer can also be specified as "maybe yes" or "maybe no" with a varying degree of certainty. An answer of "unknown" is equivalent to a "maybe yes" with a 50% degree of certainty. This concept is further discussed in the next section.

The idea of a root-cause analysis by using GTST is then simple. Starting form the top by answering questions posed to each goal, a "path" to a failure can be established by following all "failed" goals and subgoals. This is backward chaining process. Since the questions posed discrimination between "failed" and "succeeded" goals, only failed goals are further investigated which in turn saves our search domain. This process is called depth-first search. A flow chart of our depth-first search strategy for root-cause failure analysis using the GTST type knowledge-base is shown in Figure 2. As an example, Figure 3 shows a typical GTST for determining failures due to "crack".

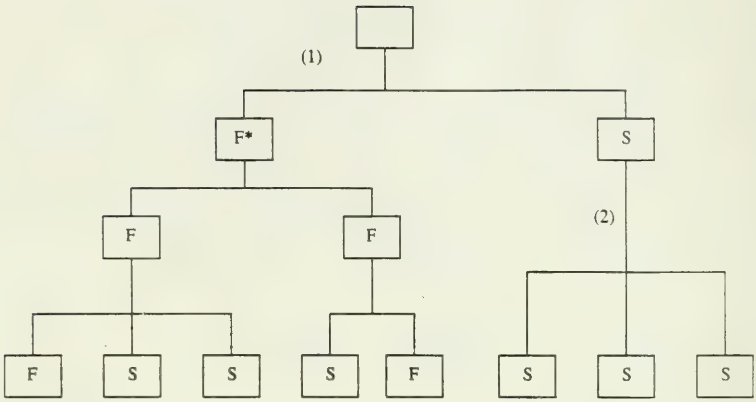
MULTI GTST MODEL -- AN EXTENSION OF GTST MODEL

In real-world applications, multiple knowledge are often needed. For example, Figure 4 hierarchically shows the relationship of certain generic specialized areas such as crack, fatigue, and corrosion to the goal of preventing failure of a turbine. If modeled, this concept can identify complex failures such as "crack induced corrosion", or "pitting in a hardware part due to impact of foreign material resulted from a wear mechanism of another part". Therefore in order to avoid repeating a specific knowledge-base for each component, and mode of failures, a multi-GTST model is developed to treat for basic failure concepts generally.

The Multi-GTST model is essentially an extension of the GTST model, where we do not put all the experts' knowledge in one huge GTST, but put more generic knowledge into a reduced sized tree. The Multi-GTST approach has allowed us to search for complex causes of failure by developing a GTST for the major plant systems or components of concern, and developing separate generic GTSTs that models the basic mechanisms by which components fail (for example cracks, corrosion, or fatigue). When one or more of these mechanisms of failure is suspected, the respective GTST will be searched to establish the hypothesis. The architecture of an expert system that works, based on this multi-GTST concept, is shown in Figure 5.

FUZZY LOGIC

It was indicated earlier that answer to the question posed for each goal to determine its status may be uncertain. In this section we will explain how the fuzzy logic method is used to deal with this problem.



* F means that the goal is failed, and S means that the goal is successful.

(1) The search process traces down the failed goals, and checks all the subgoals of the failed goals. Upon reaching the bottom, push all the failed goals to the failure list.
(2) If a goal is successful, then it is not necessary to check its subgoals.

Figure 2. The Flow of the Depth-First Search Process

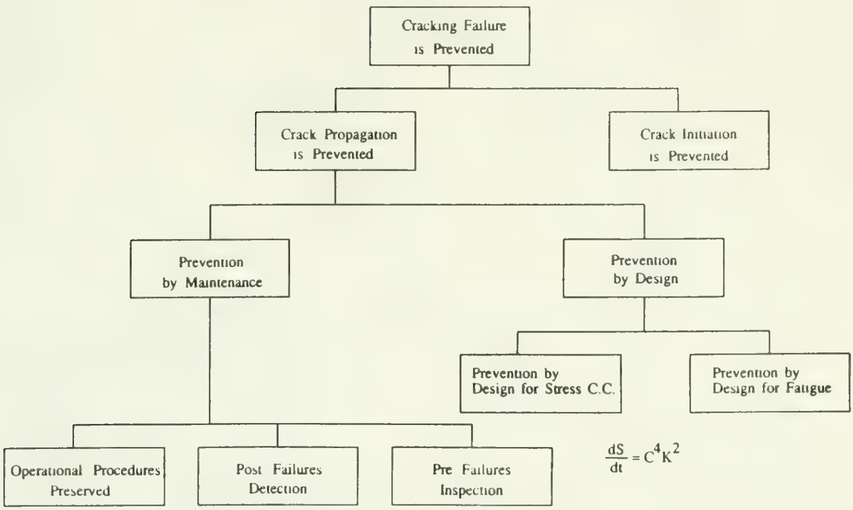
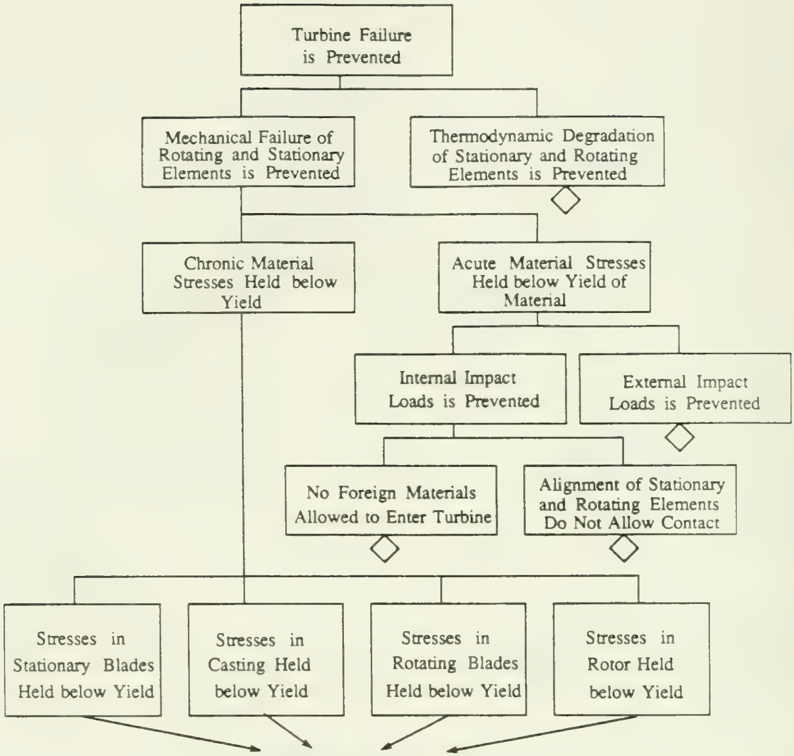


Figure 3. Crack Prevention Goal-Tree (Partial)



<u>GENERIC STRESSES</u>

. **Thermal Stresses**	**Over/Under Cooling Leading to High Thermal Gradients**
. **Mechanical Stresses**	**Speed, Annular Loading on Blades, Shock (Interstage Pressure Ratio)**
. **Erosion -**	**Interstage Water Removal via Drains, etc.**
. **Fatigue -**	**Vibration, Excitation of Blades**
. **Cracking -**	**Environment, Local Stresses, Crevice Concentration Mechanisms, Electrolyte Control**

Figure 4. Turbine Failure Prevention Goal-Tree (Partial)

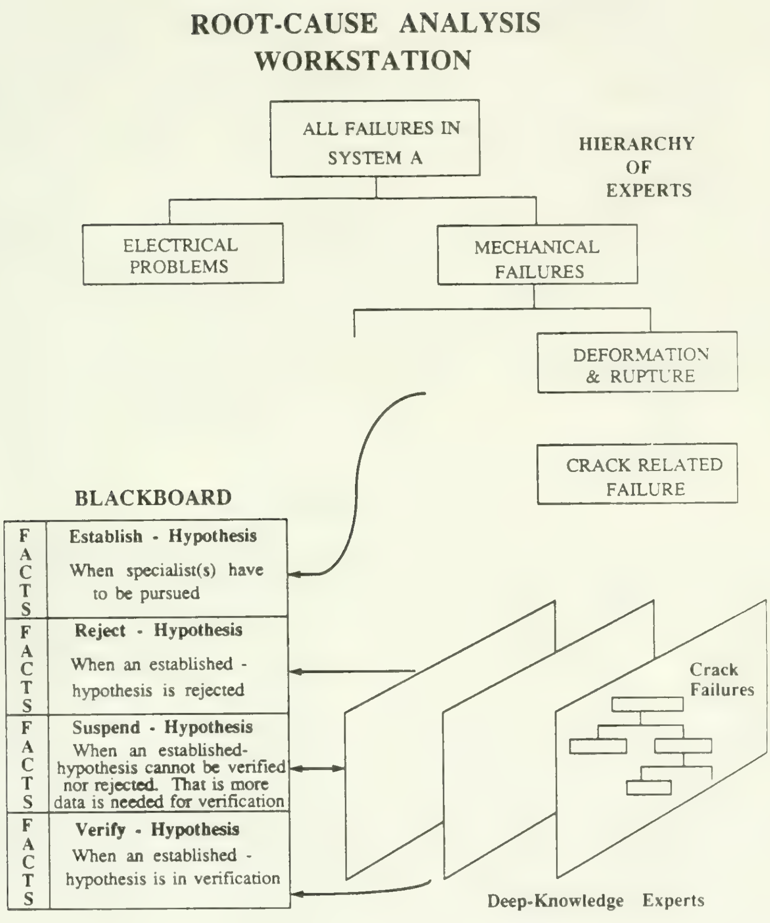


Figure 5. The Blackboard Concept in a Multi-GTST model

Some uncertainty handling methodologies, for example classical probability analysis [11] and theory of evidence [12], have been developed and used. Because the applications of these theories need failure distributions which are often unknown or unattainable in most complex process systems, in the present work fuzzy logic concept is used to handle uncertain information.

In fuzzy logic, the truth values or validities of logical propositions are intuitively considered continuous. For example in our case, a continuum of grades of success certainty value (between 0 and 100) is used to indicate the degree of our belief to the answer provided for the question which indicates the achievement of a goal. The basic algebraic operations of fuzzy logic by Zadeh [13] is defined as follows: 1) The operator "AND" is interpreted as the minimum of the truth values of the operandi. 2) The operator "OR" is then interpreted as the maximum of the truth values of the operandi.

In our application, in order to manipulate uncertainty associated with success or failure of a goal is based on fuzzy logic. According to the user's confidence on the answer to the question(s) that proves success or failure of a goal (e.g., 70% YES), the success confidence of the goal is established. If its success confidence is greater than 90%, then the goal is considered enough to be considered totally successful. Otherwise, a algorithm is used and the success certainty of the goal will be determined. If the logic of the goal is **AND** (i.e., all subgoals are required for the success of the parent goal), the success certainty of the goal is modified as the minimum of the success certainty of its subgoals. If the logic of the goal is **OR**, then the success certainty of the goal is then modified as the maximum of these success certainty.

3. IMPLEMENTATION

The whole system contains four fundamental components: 1) the Goal Tree-Success Tree models to represent experts' knowledge; 2) an expert system shell; 3) a IBM PC/AT micro computer equipped with 3 mega-bytes memory and a mouse as a pointing device; 4) and a Lisp environment -- Golden Common Lisp Version 2.2 [14]. In this section our discussion will be focused on the application of the theoretical models and the development of our system shell.

3.1 THE SYSTEM SHELL

This system has been designed to work not only in an efficient but in a user-friendly manner. Therefore the program possesses the following capabilities:

- to evaluate the nodes in the knowledge tree
- to present the predefined question to the user and acquire answer from him,
- to accept the fuzzy information from user,
- to check the consistency of the information acquired from the user,
- to display the root causes in a hierarchical structure,
- to communicate the information among GTST trees when multiple knowledge trees are required,
- to establish or modify the knowledge tree with a full screen editor,
- to present the knowledge tree by a mouse driven feature.

Basically, the system has been divided into the following subsystems, each of which will perform some specific functions in order to accomplish the requirement mentioned above:

- A knowledge-base
- an inference engine
- a communication module to collect information from multi knowledge-bases
- a user-friendly man machine interface
- a full screen editor to create or modify the knowledge-bases

In the following sub-sections, we will present a brief introduction to these subsystems.

3.1.1 Frame Structure of the GTST Knowledge Tree

In order to cover the functions listed above, we have developed an expert system shell ourselves instead of using a commercial shell which is more general purpose. First we have designed the fundamental structure to house the knowledge-base. A framed base modeling is used to represent the GTST knowledge-base. A frame is an associated list which contains several slots [15]. The structure of the frame used in our application contains the following slots:

SLOT NAME	DESCRIPTION OF THE SLOT
NAME	Name of the node which is the identifier of the frame.
PARENT	Contains the names of its parent(s) nodes.
CHILDREN	Contains the names of its children nodes.
LOGIC	Either 'OR' or 'AND'
MESSAGE	Describes how certain evidences can describe the goal.
QUESTION	A Yes/No question that is used to relate an evidence (data) with the existence of the success or failure of the goal.
ANSWER	Either YES or NO.
SUCCESS-VALUE	A success certainty value (from 0 to 100) associated with the answer slot.

3.1.2 Inference Engine

The inference engine is the heart of the whole system, in which we guide the program to search through the GTST model. Therefore, the major work of the inference engine is to apply the depth-first search described earlier in a recursive way to evaluate the nodes in the knowledge trees. In the evaluation of the nodes, the engine usually poses the user with questions in order to verify a goal failure and establish a path of the failure. The elementary aspects of this inference process is described in our earlier publications [2,3]. Because the nodes in the GTST knowledge-base are sometimes cross-linked (i.e. some of the nodes have multi parents node), some of the answers acquired from the user may conflict one another. This inference engine is also designed to check the inconsistency of the answers acquired from the users. More detailed aspects of this engine is also discussed in section 3.2.

3.1.3 Blackboard Consultant

Whenever the inference engine finishes searching one particular GTST knowledge-base, it comes out with a set of root-causes which are passed on to the communication area. This area is controlled by a module called 'blackboard', which will evaluate the root-causes and determine whether they are complete. An other reasoning and deeper investigation can still be performed via other existing GTST knowledge-bases (other specialists). In unsatisfied case, the Blackboard will generate 'hypotheses', then select and search other appropriate knowledge tree which can support the hypotheses. This process will be repeated until all the hypotheses are proven or rejected. Finally, the blackboard will have a set of solutions in a solution area which contains possible root-causes of failures.

3.1.4 Man Machine Interface

Although the man machine interface does not deal with theoretical complexities, it is a complex process that forms the front end of the system. We use the mouse-driven manual type man-machine interface with pop on menus and numerical values to select. This has simplified the usage of the system. Also, we have applied the 'multi windows' feature in the man-machine interface, which allows more diverse and concise information presentation.

3.1.5 GTST Editor

Due to the consideration of convenience for the editing and the modification to the GTST knowledge-bases, it is essential to provide a user-friendly editor. Although, the GOLDEN COMMON LISP environment that is used here does provide the editor to edit the program and the text code in a lisp environment, it is very inconvenient to edit a knowledge-base written in a lisp format which contains a lot of key words and parentheses. Therefore, we have developed an editor that allow user to edit or modify GTST in full screen that the user can move the cursor everywhere around on the screen to input the information to the knowledge-base in simple English. Whenever the knowledge-base needs modification, the user can just use the editor to edit the text of the knowledge-base (see Figure 6). The editor will convert the text in the editor into the lisp format. Furthermore, user can modify the structure of the "FRAME", which contains several slots and facets, as the system changed without additional effort to change the huge amount of the lisp keywords and pairwised parentheses.

Actually, the editor has three different modes 1) Frame base Input mode, 2) Lisp Formatting mode, and 3) Manual editor mode. The Frame base input mode is the mode in which user input the knowledge-base. In the Lisp formatting mode user can set up the frame structure including all the lisp key words and parentheses in the lisp format including the name of each slot. For convenience, the manual editor mode is designed to allow user to edit the description for each line on the screen so that the information at the right position can be input. Once the user setup the frame structure the editor will create a file to store the information that user input. Also, these information can automatically be converted to lisp language format which is applicable in the expert system.

1 NAME =	PUMP
2 PARENT =	
3 CHILDREN =	PMP-2 PMP-3 PMP-4 PMP-5
4	
5 DESCRIPTION =	Failure of Pumping System Prevented
6	
7 REFERENCE =	
8 LOGIC =	AND
9 MESSAGE =	Pump System Failure
10	
1	
2	
3 QUESTION =	Is it correct that all other parameters (other than Q and H)
4	are in the limits and Prime Mover Power matches Water
5	Power in the vicinity of BEP?
6	
7	
8	
9 ANSWER =	YES
20 PATH =	(1)
1	

Insert **Adding Page** **Top Unchanged** **> Editing...**

1:Find 2:Edit 3:LDIsp 4:LDfrm 5:LDdef 6:SVIsp 7:EDIft 8:EDIsp 9:Save 0:Quit

Figure 6. Full Screen GTST Editor

3.2 SYSTEM STRUCTURE

The Top-Design Software design technique is applied during the development of the system. Here we have described the system structure in a Top Down manner in order to provide a systematic overview to the system.

The topmost module (called MAIN-MANUAL) of the system is a module which accepts the choice from the user through the mouse-driven manual selection. It is the entry to the system in which there is only a simplest loop (see Figure 7,8) to perform the function selected by the user through the "mouse". A flexible table of manual, TOP-MENU, is designed for the purpose that any additional subsystem can be added on simply by adding additional items to TOP-MENU without having any change to the execution code. The TOP-MENU is actually a list of several items which are made of a string of text associated with a function list. The function list is the name of the subsystem which can be called by this top module. Currently, the manual contains the items as the following:

- DIAGNOSIS	To perform root-cause determination process where will ask the user several question in order to establish causes of failures.
- ROOT-CAUSES	To provide the user ability to look at the hierarchical structure of the root causes.
- POSSIBLE-CAUSES	To provide the user ability to look at all the other possible causes with lower certainty.
- SHOW-TREE	To display the hierarchical tree structure of the knowledge-bases in a mouse driven manual, in which the user can move the mouse to see the contents of the node of interest.
- EDIT-TREE	To provide the user to edit or modify the knowledge-base.
- LOAD-TREE	To provide the use to update the knowledge tree which is already loaded in the system, when the knowledge-base has been modified.

3.2.1 Diagnosis Process

The subsystem is the portion of the system which contains the Multi GTST model. Namely, this subsystem contains the inference engine and the Blackboard Consultant as mentioned above. The procedure of the diagnosis can be described as below:

- Accept the problem issued by the user and load an appropriate knowledge tree into the system,
- Search the knowledge tree from the very top of the tree, during the recursive search process the subsystem will perform the following tasks:
- Acquire information from the user by asking proper Yes/No questions associated with the goals in the tree, which is designed in a pop-out window so that the user can easily look at the question and choose the answer either Yes or No, including degree of uncertainty with the answer,
- Check the consistency of the information acquired from the user,
- Find out the root causes to the problem issued by the user,
- Find out the less likely root causes associated with small confidence value (lower than 80%).
- Determined whether the outcoming root causes is sufficiently satisfying or not, and generate some hypotheses which need to be support by some other information,

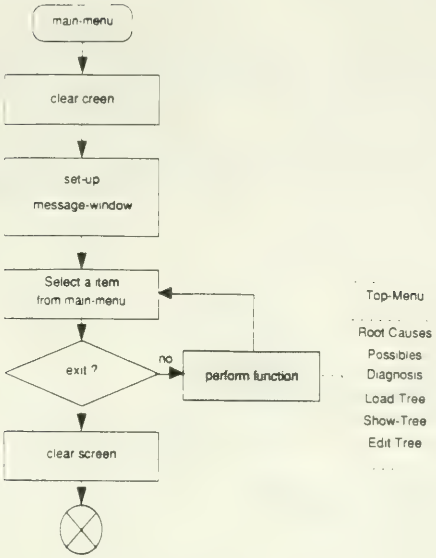


Figure 7. Main Entry of the System

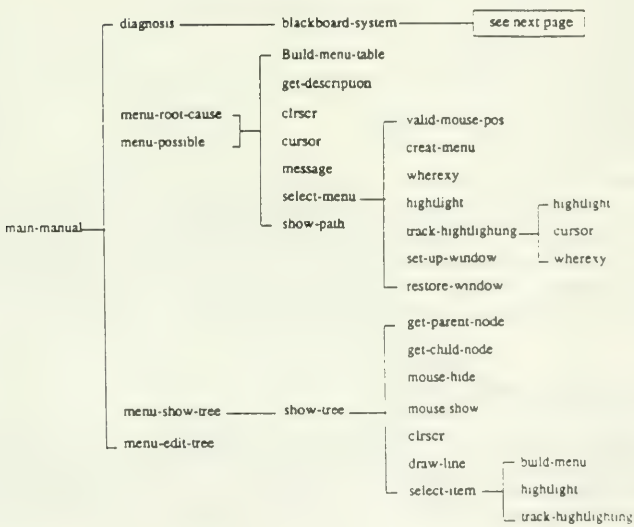


Figure 8. The Hierarchical Structure of the Program-1

- Load another knowledge tree to search further information to support the unsatisfaction of the root causes.

The above functions are accomplished by two separated modules (see Figure 9.10), one is called the TREE-SEARCH the other is called BLACKBOARD-CONSULT. The inference engine, which is embedded in the module of TREE-SEARCH, evaluates the success value of each node/goal of the knowledge tree. This inference engine, called EVALUATE, basically evaluates only the siblings and the children of a node, however, by recursive calling to this engine itself, the whole knowledge tree can be searched until the bottom/leaf node is reached, and the root causes can be found. The following pseudo code shows how the inference engine works:

```
Ask the success value from the user;
if Successful then,
    set all its children nodes as successful nodes,
;otherwise,
    fetch its children nodes
    if the children nodes exists then,
        evaluate all the children nodes,
        make the success value consistent with the evaluation,
    ;otherwise,
        consider this node as the bottom,
        remember it as a root-cause,
;fetch its sibling nodes,
if the sibling nodes exists then,
    if the logic gate of their parent are AND-gate then,
        evaluate the sibling nodes,
        return the minimum success certainty value of this level,
    ;otherwise,
        evaluate the sibling nodes,
        return the maximum success certainty value of this level,
;otherwise,
    return the success certainty value,
```

Once the root causes are determined for one specific knowledge tree, the root causes will be searched whether there are more relevant information that can be found in other GTST knowledge-bases. The system will generate the hypotheses based the searching of the root causes , then it will load and search the appropriate knowledge tree one by one until the hypotheses are all proven. Finally the system will generate two final lists, one is the partial solutions and the other is the possible solutions.

3.2.2 Window to the Root Causes Found

When the root-cause problems are determined, the two functions (MENU-ROOT-CAUSES and MENU-POSSIBLE) will allow the user to look at the reasoning that the system has selected in a manual-like form. The user can use the mouse to select any of the possible root-cause and click the mouse to see the hierarchical reasoning structure of any of the root causes.

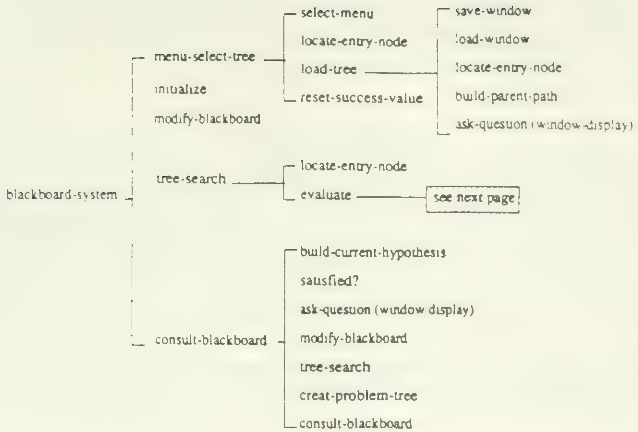


Figure 9. The Hierarchical Structure of the Program -2

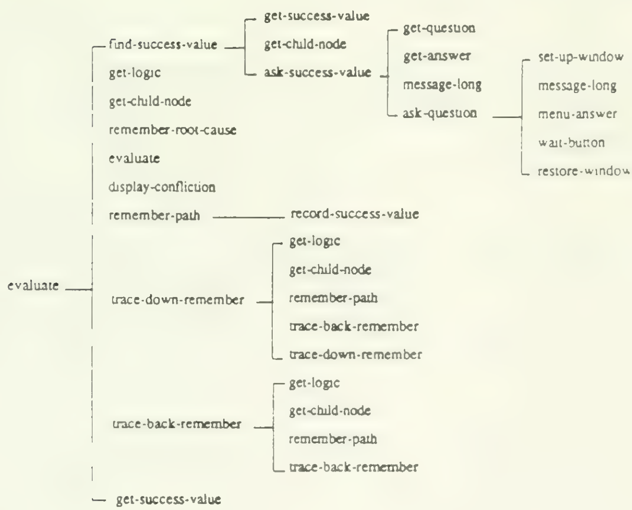


Figure 10. The Hierarchical Structure of the Program-3

First, the window will show a list of the root causes found via the diagnosis process. The user then can move the mouse to select the item of the root causes he want to further look. A hierarchical structure of the selected root cause will pop out so that the user can have systematic idea how the root cause propagate to the observed failures and other evidences gathered. That is how everything fits together.

4. CONCLUSION

In this paper an approach for the integration of several GTST models in analysis of the root-causes of system failures has been presented. Currently, a prototype expert system has been developed and provided with several generic GTST knowledge-bases for major systems such as nuclear plant feed pumps, ways a C-E PWR can scram as well as fundamental mechanisms of failures including corrosion, crack, and fatigue. Some experimental runs have been tested, and have disclosed several advantages in using this research.

- A generic GTST knowledge-base, which consists of a detailed knowledge of a specialized domain, is activated only when its related information is required. Therefore, the computer memory load and system search time can be substantially decreased.
- For assessment of the worth of improvements to problem equipment, it may subject to change in process systems. However, in the present expert system only the correlated knowledge GTST model needs to be modified. That is, Multi-GTST model offer a better maintenance ability.
- Since the expert system contains different knowledge-bases and provides a promising explanation facility, a system analyst can put more insight to evaluate the performance of hardware components.

ACKNOWLEDGEMENTS

This research work is supported by Maryland Industrial Partnerships (MIPS), and by the Baltimore Gas and Electric Company.

REFERENCES

1. R. N. Hunt and M. Modarres. "Integrated Economic Risk Management in a Nuclear Power Plant, 1984 Annual Meeting, The Society for Risk Analysis, Oct. 1984.

2. M. Modarres, et al. "Development of A Root-Cause Analysis Workstation and Its Applications", 15th Interim Conference, Portland, June 1988.

3. D. T. Chung and M. Modarres. "GOTRES: An Expert System for Fault Detection and Analysis," Reliability Engineering and System Safety, 24, 1989, pp. 113-137.

4. I. S. Kim and M. Modarres. "Application of Goal Tree-Success Tree Model as the Knowledge-Base of Operator Advisory Systems," Nuclear Engineering and Design, Vol. 104, 1987, pp. 67-81.

5. L. D. Erman, et al. "The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," Computer Surveys, 12(2):213-253, 1980

6. M. A. Kramer and B. L. Palowitch Jr. "A Rule-Based Approach to Fault Diagnosis Using the Signed Directed Graph," A.I.Ch.E. J. Vol. 33, No. 7, July 1987.

7. J. M. Douglas. "A Hierarchical Decision Procedure for Process Synthesis," A.I.Ch.E. J., Vol. 31, 353, 1985.

8. J. Rasmussen. "The Role of Hierarchical Knowledge Representation in Decisionmaking and System Management," IEEE Transactions on Systems, Man, and Cyber., Vol. SMC-15, No. 2, 1985.

9. H. A. Simon. "The Architecture of Complexity," Proceedings of the American Philosophical Society, Vol. 106, No. 6, 1962.

10. G. J. Birky, T. J. McAvoy and M. Modarres. "An Expert System for Distillation Control Design," Computers and Chemical Engineering, Vol. 12, No. 9/10, 1988.

11. A. J. Palay. Searching with Probabilities, Pitman, Inc.,1985.

12. G. Shafer. "A Mathematical Theory of Evidence," Princeton University Press, Princeton, NJ., 1976

13. L. A. Zadeh. "Fuzzy Sets," Information and Control, 8, 338-353, 1965.

14. Golden Common Lisp 286, GCLISP-286, Gold Hill Computers, Inc., Cambridge, MA, October, 1986.

15. R. Fikes and T. Kehler. "The Role of Frame-Based Representation in Reasoning," Communications of the ACM, Vol. 28, No. 9, pp. 904-920, Sept. 1985.

Substation Design Using CAD and Expert Systems Tools

JEAN-MARC PELLETIER
Institut de Recherche d'Hydro-Quebec (IREQ)
1800 Montee Ste. Julie
Varennes, Quebec, Canada J0L 2P0

ROGER BEAUCHEMIN
Hydro-Quebec
855 Ste. Catherine Est
Montreal, Quebec, Canada H2L 4P5

ABSTRACT

Substation design is, nowadays, a graphics oriented task with the designer using CAD software tools; using these tools, modifications are easily made to the design if required. However, CAD tools are not intelligent and cannot verify the design itself except if the rules of verification are imbedded in the CAD environment. On the other hand, expert system tools are well suited to provide diagnostics of problems and to verify that the rules have been followed. If we could combine these tools, intelligent CAD environments could be created to assist the substation designer in his task.

In this paper, we present the work that has been done to merge these two environments in order to provide the substation designer with a useful tool; we present the constraints that were imposed in the design, the methodology chosen to achieve a practical and flexible tool, the problems and solutions found during development.

INTRODUCTION

Substation design is a process by which a designer gathers elements to provide transformation of voltage levels and/or to switch line and load circuits. The elements that are used in the design of a substation are:

- power transformers

- circuit breakers
- disconnect switches
- voltage and current transformers for protection purposes
- inductors
- capacitors
- gates, fences and service roads

Designing a substation is a rather simple task when one looks at the final layout: the designer has merely to connect the power transformers (through their circuit breakers and disconnect switches) to the power lines that carry power in; the low voltage side of the power transformers are connected to the power lines that carry power out. Units such as voltage and current transformers are used to protect power transformers from severe operating conditions; inductors and capacitors are used to correct the power factor and, obviously, gates and fences to protect unauthorized access to the site. However, the tedious part of designing a substation is verifying that all elements have been correctly connected, that the clearances between the elements have been respected, that the noise levels are within reasonable limits,...

Substation design of low voltage substations (25 kV, 120 kV and 315 kV) is a mature activity and does not involve any new design criteria; the elements required for this type of design are known, their interconnections and the clearances between the elements are standardized within the utility. Henceforth, for this class of substation, junior engineers can be responsible for the design of the substation and the rules to follow are well known and documented. In this case, the designer has to follow a set of guidelines and rules well established from past experience; the final layout can then be given to an expert for final verification. This is a routine task that could well be automated.

On the other hand, substation design of very high voltage substations (735 kV and above) is a task that is usually given to an expert; design criteria are not well established, rules of thumb oftenly applies in this kind of design, optimization of the design has yet to be done. For all these reasons, this process is not clearly defined and has to be refined before any automation can be done.

ADVENT OF COMPUTERIZED TOOLS

In this section, we present a brief historical overview of the way that substation design was done in the past; we then present the technologies that are used today

and introduce new technologies that can still enhance the design automation of substations.

Historical viewpoint of substation design

Before the advent of individual computerized workstations, designing a substation required at least two individuals: a designer (whether an expert or a junior engineer) to carry out the design and verification of the substation layout and a draftsman to actually draw this layout on paper. Modifications and changes to the layout oftenly involved a complete redraw, resulting in the final version of the substation. From this description, it is clear that designing a substation required a long time and many resources were needed.

The introduction of workstations and personal computers in the offices and the introduction of related softwares has given a big boost to productivity; specifically, the introduction of CAD software and it's application to substation design has lowered the time required to perform this task. However, the introduction of this new tool, although lowering the design time, still requires that two individuals team up on a design: a designer and a draftsman. In this case, the draftsman has merely replaced his pen and paper for a computer display, which improves his efficiency, but the human resources are still the same. A major improvement, at this point, would be to lower the number of persons required to design a substation.

Expert systems and expert system shells

Expert systems are computer programs built using expert system shells; they differ from computer programs written in conventional languages (such as PASCAL, FORTRAN or C) from the fact that the algorithm (rules) are kept outside the main program and kept inside a knowledge base. An expert system is made of two elements:

- a knowledge base where the rules are inserted
- an inference engine which links and executes the rules within the knowledge base

An expert system differs from a program written in a conventional language for these two reasons:

- the addition of rules and/or the modification of relationships between elements can be easily made without program recompilation
- an explanation facility is provided with the expert system whereby the user can request an explanation of the deductions made by the expert system

Using an expert system shell, the developer can build an expert system according to the procedures followed by this kind of development.

Although expert systems and expert system shells are new technologies in the computer area, they have proved to be very powerful tools to build and maintain knowledge; they have been used in different domains (medecine, business, industries,...) and could also be used in substation design.

NATURE AND USEFULNESS OF PROJECT

The project, as defined by our client, consisted in a prospective evaluation of the use of expert systems applied to the design of substations; given the constraints that will be listed later in this paper, an expert system had to be built to verify the design of a well known class of substation, namely a 120 kV / 25 kV substation with two power transformers, the design being expandable to four power transformers. The only rule required in the original specification was to check the clearances between the elements of the substation; additionnal rules could be added during development if necessary.

This integrated tool, if the development is successfull and if the final product is well embedded into the designer's environment, could have significant impacts on utilities designing substations: not only could the human resources be reduced to a single individual designing the substation with less knowledge of the design to be carried out but a more complete verification of the design would be performed by the system, freeing the designer from simple and repetitive tasks.

Given the novelty of such an application, the main goal was to evaluate the technology as it exist today and the client did not expect a working product at the end of the project, since the risks of failure were important.

Finally, at the completion of this project, the appropriate recommandations will be made as to whether a useful product should be developped in the future.

HOW TO SETUP AN INTELLIGENT CAD ENVIRONMENT

The next section of this paper presents different ways of adding rules verification within a CAD environment; two possible solutions are presented as well as a derivative of the second solution which we have retained.

Embedding rules within the CAD environment itself

Many CAD softwares give the user the opportunity to write external functions (in some form of language usable by the CAD software) to perform repetitive actions or to implement new functions using the predefined functions; as an example, Autocad enables the user to write functions in Autolisp which are interpreted and executed by Autocad. Accordingly, the designer could implement the rules,using this language, within the CAD environment itself and an intelligent CAD environment could be built.

This solution has many advantages:

- there is no need to call external programs to verify the design
- there is no need to pass data to external programs
- the execution of the verification process is faster
- the end user always remains inside the CAD environment

This solution shows some disadvantages:

- the developer has to learn the language in which the rules will be coded
- the rules are imbedded inside the functions, which is equivalent to program the rules using conventional languages
- adding or changing rules, as the number of rules increases, makes a large system unmanageable
- the end user, when the system is released for operation, has to learn the language if he wishes to add or modify rules
- the utility relies on the developer to add or modify rules if the end user is unable to do so
- there is no builtin explanation facility

Although this approach has many advantages, it cannot be retained for systems where a large number of rules is required, as is the case for substation design. For smaller systems, given the disadvantages, this approach could be used.

Combination of CAD environment and external expert system

The use of an expert system, as mentioned previously, is well suited for verification purposes; as such, the rules can be added to the knowledge base and maintained by the expert system shell. Figure 1 presents a graphical view of this concept.

The CAD environment is the graphics environment used by the substation designer; this environment is not modified by this new concept and the facilities developed by the user (such as functions or graphical symbols defined by the user) remain the same. Whithin the CAD environment itself, special functions can be specifically written to extract data from the drawing and write this data to the output file; then, a call to the operating system is made to execute the external expert system which reads this data and validates the rules. The output of the expert system is written to it's output file and then read back by another function in the CAD environment. This last function decodes the output of the expert system and is able to insert or move elements, according to the results of the expert system.

All the elements presented in figure 1 represent the new "integrated CAD substation design environment" as we define it and the user is unaware of the internals of this concept; as a user, he always remains in the CAD environment itself and doesn't see any difference from his previous way of operation.

This solution is more complex than the previous one but has many advantages:

- the rules are maintained inside the knowledge base
- the rules are not coded inside the program
- adding or modifying rules is simple even if the expert system becomes large
- using the expert system shell, a novice can add rules easily

However, it has some disadvantages:

- the expert system is an external program called from the CAD environment
- special functions to pass data and read data from this external program have to be written

- the overall performance of the whole system is slower
- in some cases, due to operating system's constraints, it can be difficult to call this external program from the CAD environment

Given all the advantages and disadvantages of this solution, our client requested that we implement this second solution; although more complex, it will provide him with much greater flexibility for future updates.

Final solution retained

The final solution that has been retained is derived from the previous concept and is presented in figure 2.

Let's now examine the differences between this solution and the previous one. Since the rules that had to be verified were related to the presence of elements as well as the clearance between these elements, we did not want to "hard code" any constant data inside the knowledge base (enabling the user to modify the values of clearances without modifying the rules themselves); also, since some data is common to both the CAD environment and the expert system, modifying this common data would have been impractical. Consequently, the file "cad_sys.dat" which houses the values of clearances (and other information) is shared between both tools; this file is only readable and is not modified by any of these two programs.

Another difference between the retained solution and the previous solution is the inclusion of the "startup expert system". This expert system, executed outside the "integrated CAD substation design environment", enables the user to specify paramaters before any drawing is made; these parameters are then used by the CAD environment to establish a startup drawing. This tool is an addon and gives the user more flexibility when designing substations.

DESIGN CONSTRAINTS AND CHOICES MADE

Since the expert system had to be integrated to an existing CAD tool in use in more than 10 sites, some specific hardware and software constraints were imposed by the client; these constraints are listed in this section as well as the choice of the expert system shell used for development.

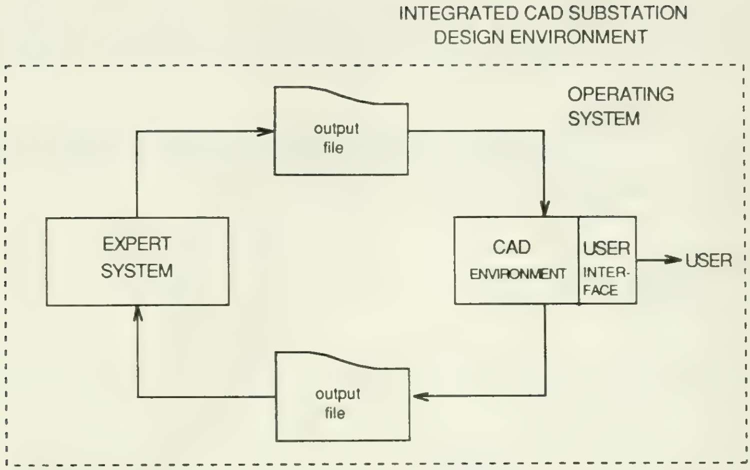


Figure 1. Combined CAD and expert system environment

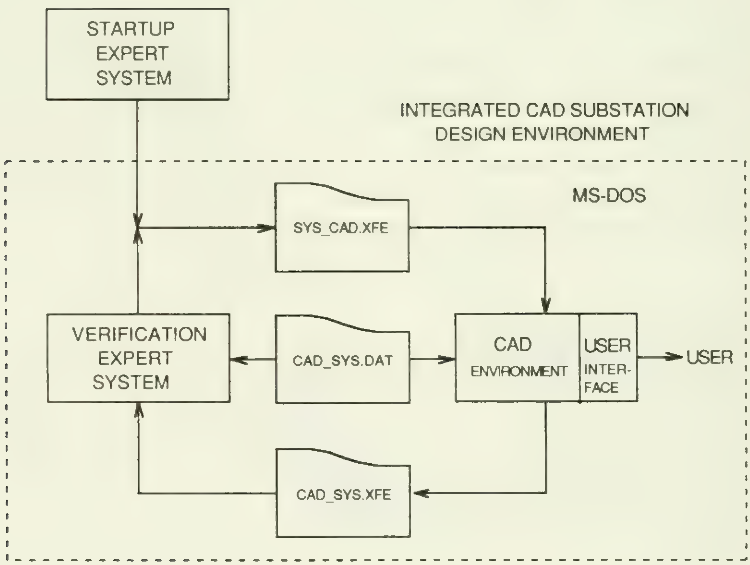


Figure 2. Configuration adopted for this project

Constraints imposed by the client

Among the constraints imposed by the client, the following were the most important:

- HARDWARE : the "integrated CAD substation design environment" had to work on the IBM PC class of hardware
- SOFTWARE : the CAD tool used by the client is Autocad release 9.0
- OPERATING SYSTEM : the operating system in use by the client is MS-DOS

These constraints seemed reasonable, at the time the project was started, and future work will show that some of these constraints were major constraints that could impair the sucessful development of the whole system.

Choice of the expert system shell

As a developer, we had full flexibility as to the choice of the expert system shell; past experience with the chosen shell as well as it's integration in the "integrated CAD substation design environment" lead to the choice of Rulemaster release 2.0. Specific reasons for this choice are:

- this shell was well known to the developer and no time was required to study a new tool
- this shell can manage a large number of rules and makes future expansion possible
- this shell translates the rules in programs written in C, which are then compiled into a "binary executable file" (a .exe file); this file can easily be called from the CAD environment

BUILDING THE "INTEGRATED CAD SUBSTATION DESIGN ENVIRONMENT"

Designing an "integrated CAD substation design environment" when limited experience has been assessed in the past is a challenging task. This section presents the formats chosen for data exchange; then a list of the functions added to the CAD environment to deal with external programs (the verification and startup expert systems) as well as the verification expert system designed to verify the rules is made.

Formats of data files

The data files needed to implement the "integrated CAD substation design environment" are explained below. The formats chosen were optimized for the rules which had to be coded and could be modified in future releases.

cad_sys.dat file. This file houses the data that is shared by both the functions inside Autocad and the verification expert system. Table 1 lists the format of this file.

As mentioned before, the purpose of this file is to hold block names which will be analyzed, to hold constant values such as clearances between elements, default values for operational parameters, ...

The first field is a key field and identifies the variable name or the block name being analyzed; the second field identifies the nature of the first field (a variable or a block). The third to the eight field represents constant data applied either to the variable or the block name; and the ninth field, used for blocks, represents a reference block name.

The format of a line applied to a variable has the following structure:

- the first field identifies the variable name
- the second field identifies the first field as a variable
- the third field holds the value of the variable name
- the remaining fields are ignored

The format of a line applied to a block has this structure:

- the first field holds the block name
- the second field identifies the first field as a block
- the third and forth fields hold the typical values of x and y clearances of this block when referenced to the block identified in the ninth field (the reference block)
- the fifth and sixth fields hold the minimum values of x and y clearances of this block when referenced to the block in the ninth field
- the seventh and eight fields hold the maximum values of x and y clearances

- the ninth field holds the block name which is the reference for the block being defined in the first field

As an example, let's examine the contents of lines 2 and 7 of table 1: line 2 states that the variable "message" is a variable and has the value "oui" (yes in french): this variable being set to "yes", if the draftsman doesn't change it, it's value will be copied in the "cad_sys.xfe" file and, consequently, the expert system will send comments in it's output file "sys_cad.xfe". On the other hand, line 7 states that the block named "LC_TT" is a block and has a typical clearance, in the x direction, of 0 metres from block "STRU120", a typical y clearance of 21.410 metres from block "STRU120", a minimum x clearance of 0 metres,...

<u>cad_sys.xfe.</u> This file holds the data that is extracted from the drawing and pertains to the actual values of the operational parameters as set by the draftsman as well as the absolute values of x and y coordinates of blocks. Table 2 lists the format of this file.

The meaning of each line is similar to the one described previously but the x and y coordinates of the blocks are absolute values as opposed to the values in the "cad_sys.dat" file which are relative values.

<u>sys_cad.xfe.</u> This file is the output of the expert systems (either the verification expert system or the startup expert system) and table 3 presents it's format.

The output of the verification expert system, as will be explained later, has two different values: each line can either hold a "comment" (in case the draftsman has set the message variable to "yes") identified by the keyword "COMMENT" in the first field followed by the comment itself or an action followed by the type of action, the block on which the action applies and finally, the new x and y coordinates of this block.

<u>Functions added to the CAD environment</u>

Two functions were written in Autolisp to take care of the exchange of data with the external programs (expert systems); these functions are "verification.lsp" and "anal_res.lsp" and a schematic representation of their duties within the CAD environment is presented in figure 3.

verification.lsp. This Autolisp function scans the file "cad_sys.dat" and, for each block name identified in the first field (refering to table 1), if the block is drawn, it's absolute x and y coordinates will be written in the file "cad_sys.xfe"; secondly, the actual operational parameters as set by the draftsman (or their default values) will be written to the same file.

anal_res.lsp. This function scans the file "sys_cad.xfe" (the output of the expert systems) and executes the actions listed in the file or sends the comments to the draftsman; as an example, refering to line 6 of table 3, this function will move the block named "CHEM_ACC" to new coordinates.

Verification expert system

The verification expert system executes two rules in five steps; the rules that are verified are a rule that checks the presence of elements in the drawing (is the design complete?) and a rule that checks the clearances between the elements (are the clearances between elements at their typical value, below their minimum value,...?). Depending upon the settings of operational parameters, it's output (to the file "sys_cad.xfe") can be:

- COMMENTS if the draftsman has set the message variable to "yes"
- ACTIONS such as INSERT if the auto_insert variable has been set to "yes" and if an element is missing
- ACTIONS such as MOVE if the auto_move variable is set and the element is not properly positioned

Figure 4 summarizes the steps executed by this expert system as each step will be analyzed separately.

Building the graph of dependencies. Refering to table 1 and to the explanations of the format in the file "cad_sys.dat", it was mentioned that every block that is defined in this file is a block whose clearances are related to a reference, the ninth field of the line. When the expert, who establishes the clearances between the elements, inserts these values in the "cad_sys.dat" file, he can define every element in respect to a previously defined element or in an unorderly manner. For example, let's assume that element A is referenced to element B which is also

Table 1

LIST AND FORMAT OF FILE "CAD_SYS.DAT"

CLE	QUAL	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	REF
message	VAR	oui						
auto_move	VAR	non						
auto_insert	VAR	non						
REF	BLOC	0.000	0.000	0.000	0.000	0.000	0.000	REF
STRU120	BLOC	0.000	0.000	0.000	0.000	0.000	0.000	REF
LC_TT	BLOC	0.000	21410.000	0.000	20910.000	0.000	25000.000	STRU120
LC_TP	BLOC	0.000	-5000.000	0.000	-5000.000	0.000	-6000.000	STRU120
CHEM_TP	BLOC	0.000	-7325.000	0.000	-7300.000	0.000	-9500.000	LC_TP
CHEM_ACC	BLOC	49200.000	0.000	49200.000	0.000	60000.000	0.000	CHEM_TP
LC_IM	BLOC	0.000	-4575.000	0.000	-4550.000	0.000	-6500.000	CHEM_TP
STRU25	BLOC	0.000	-11300.000	0.000	-11300.000	0.000	-13500.000	LC_IM
LC_DEP	BLOC	0.000	-3400.000	0.000	-3400.000	0.000	-3400.000	STRU25
STRU_CON	BLOC	0.000	-9000.000	0.000	-9000.000	0.000	-11000.000	LC_DEP
BATIMENT	BLOC	0.000	14.000	0.000	0.000	0.000	50.000	STRU120

Table 2

LIST AND FORMAT OF FILE "CAD_SYS.XFE"

CLE	QUAL	DATA 1	DATA 2
message	VAR	oui	
auto_move	VAR	oui	
auto_insert	VAR	non	
REF	BLOC	0.000	0.000
STRU120	BLOC	0.000	0.000
LC_TT	BLOC	0.000	21409.800
LC_TP	BLOC	0.000	-5000.000
CHEM_TP	BLOC	0.000	-12325.198
CHEM_ACC	BLOC	-49000.000	-12325.000

Table 3

LIST AND FORMAT OF FILE "SYS_CAD.XFE"

```
COMMENT    Element STRU120 est a la distance y normalisee
COMMENT    Element LC_TT est dans les limites comprises entre la valeur y minimale et y maximale
COMMENT    Element LC_TP est a la distance y normalisee
COMMENT    Element CHEM_TP est dans les limites comprises entre la valeur y minimale et y maximale
COMMENT    Element CHEM_ACC est a une distance superieure a la valeur y maximale
ACTION     MOVE        CHEM_ACC        -49000.000   -12325.2
COMMENT    ATTENTION : VERIFICATION FINALE DE LA REGLE DU NOMBRE D'ELEMENTS
COMMENT    Element LC_IM n'est pas present dans la conception du poste
COMMENT    Element STRU25 n'est pas present dans la conception du poste
COMMENT    Element LC_DEP n'est pas present dans la conception du poste
COMMENT    Element STRU_CON n'est pas present dans la conception du poste
COMMENT    Element BATIMENT n'est pas present dans la conception du poste
COMMENT    ATTENTION : UNE ERREUR DANS LE LA REGLE DU NOMBRE D'ELEMENTS EST TOUJOURS PRESENTE
COMMENT    ATTENTION : ANALYSE NON COMPLETE...VERIFIER MANUELLEMENT LA PRESENCE DES ELEMENTS
COMMENT    ATTENTION : VERIFICATION FINALE DES REGLES DE DISTANCE
COMMENT    Element STRU120 est a la distance y normalisee
COMMENT    Element LC_TT est dans les limites comprises entre la valeur y minimale et y maximale
COMMENT    Element LC_TP est a la distance y normalisee
COMMENT    Element CHEM_TP est dans les limites comprises entre la valeur y minimale et y maximale
COMMENT    Element CHEM_ACC est a une distance superieure a la valeur y maximale
COMMENT    ATTENTION : UNE ERREUR DANS LES REGLES DE DISTANCE EST TOUJOURS PRESENTE
COMMENT    ATTENTION : ANALYSE NON COMPLETE...VERIFIER MANUELLEMENT LES DISTANCES
```

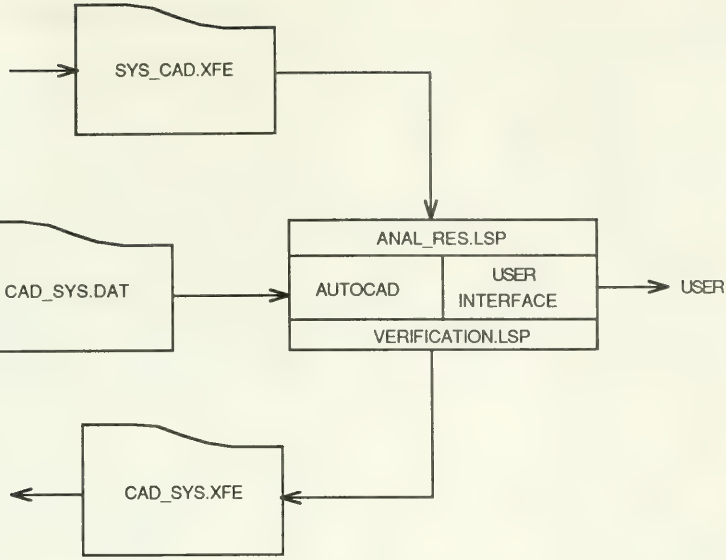


Figure 3. Inside functions of CAD environment

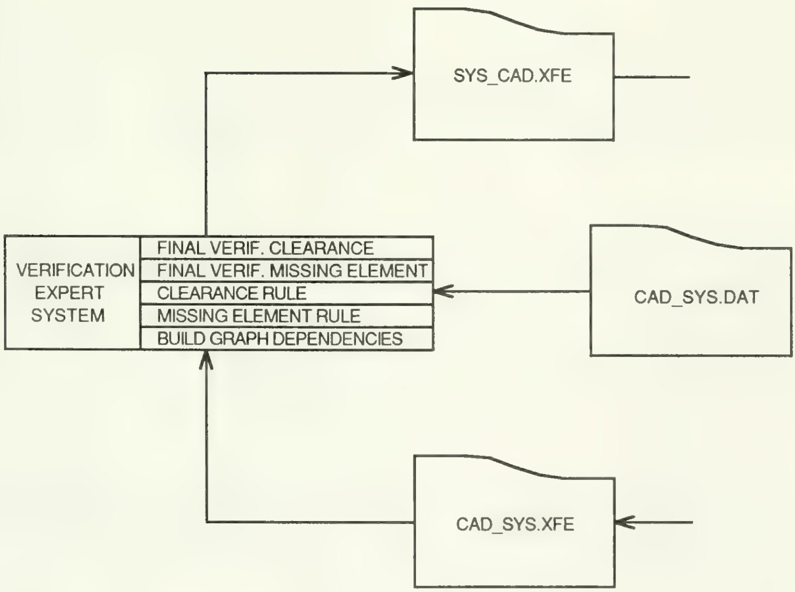


Figure 4. Sequence of verification of the expert system

referenced to element C and that the order in which these have been defined in the file "cad_sys.dat" is:

```
B    BLOC    0.000    12.000    ......  C
A    BLOC    0.000     6.000    ......  B
```

If we first check the clearance of element B in respect to element C and find the clearance to be right and then check the clearance of element A in respect to element B and find this clearance to be wrong, we cannot move element B to satisfy this second relation since this would invalidate the first relation (clearance of element B in respect to element C) which has already been checked and found to be good. In this case, the only solution would be to move element A to validate the second relation.

From this example, one can see that if the file "cad_sys.dat" has not been properly ordered, false conclusions and wrong actions could be taken by the expert system. To avoid such a possibility, we have decided to construct a graph of dependencies which translates the relationships between elements before any insertion of new elements or displacement of existing element is made; we thus infer the positional relations between the elements.

From the data listed in table 1, the graph of dependencies would be:

```
                      LC_TT    > STRU120  > REF
           CHEM_ACC > CHEM_TP  > LC_TP    > STRU120
STRU_CON > LC_DEP   > STRU25   > LC_IM    > CHEM_TP
                                 BATIMENT > STRU120
```

This graph represents the dependencies of the elements that are listed in table 1 and dependency 1 states that block LC_TT depends upon the proper position of block STRU120 which in turn depends on block REF; the other dependencies state the relations between the other elements in the design which cannot be linked to the first dependency. Evaluating each dependency separately from right to left will provide an orderly insertion of missing elements or displacement of mispositioned elements.

<u>Checking for missing elements</u>. This rule uses the graph of dependencies and verifies that each element defined in the graph is listed in the file "cad_sys.xfe" (which holds the positions of the elements currently in the drawing); three conclusions and two actions are derived from this rule:

- if the element is present, no comments are given and actions taken
- if the element is missing and if the message mode is enabled, then a comment referring to this missing element is written in the file "sys_cad.xfe"
- if the element is missing and if the auto_insert mode is enabled, an action is written in the output file with the coordinates of the element to be inserted; the file "cad_sys.xfe" is updated to reflect the inclusion of this new element

Checking clearances between elements. This rule uses the graph of dependencies and verifies that the clearances between elements are right; typical (minimal and maximal) values are read from the file "cad_sys.dat" and compared with the values in the file "cad_sys.xfe". Four conclusions and two actions are derived from this rule:

- if the clearance is right and if the user has disabled the message option, no comment or action
- if the clearance is right and if the message option is enabled, a comment is written in the output file
- if the clearance is wrong and if the user has selected the message option, a comment is written in the output file
- if the clearance is wrong and if the user has selected the auto_move option, an action is written in the output file with the corrected coordinates of the faulty element; the file "cad_sys.xfe" is updated to reflect the new position of this block

Final verification of missing elements. One requirement from the client requested exact diagnostics from the expert system; for this reason, should the verification of missing element rule fail, this rule, which scans the file "cad_sys.dat" (instead of the graph of dependencies) and the updated file "cad_sys.xfe" will issue a warning in the output file; no action is taken to insert the missing block.

Final verification of clearances. For the same reason as mentioned previously, this rule scans the file "cad_sys.dat" and the updated file "cad_sys.xfe" and will issue a warning if an element is still misplaced; no action is taken to correct the clearance.

OUTPUT OF THE VERIFICATION EXPERT SYSTEM

Table 3 lists the conclusions derived from the clearances defined in table 1 and the actual positions of the elements as defined in table 2; table 3 is thus a typical output of the verification expert system. Refering to table 3, we will comment the conclusions:

- conclusion 1 states that block STRU120 is at the typical y clearance
- conclusion 2 states that block LC_TT is not at the typical clearance but within the minimum and maximum y clearances
- conclusion 5 states that element CHEM_ACC has a clearance greater than the maximum value
- hence, since the auto_move mode is on (refering to line 3 of table 2), conclusion 6 states that an action of the type "MOVE" should be taken on the block named CHEM_ACC and that this block should be positioned to these new coordinates
- since the check of missing elements is still to be done (explained later), conclusions 7 to 14 state the final check of missing elements and mention that elements LC_IM, STRU25, LC_DEP,... are still missing in the design; specifically, conclusions 13 and 14 state that "an error in the missing element rule is still present" and that "a manual verification of the presence of elements should be done"
- conclusions 15 to 22 state the final check of the rule of clearances and mention that an error is still present even if element CHEM_ACC has been moved to it's new position; this behaviour is due to rounding errors (the y coordinate of CHEM_ACC as computed in conclusion 6 should be -12325.198 instead of -12325.2). Nevertheless, the final verification of clearances has detected this error and given the appropriate warning to the designer

At this time, we feel that the conclusions of this expert system are right and we are satisfied with it's behaviour; we feel that adding the final verification of both missing elements and clearances gives the user the assurance that no unnoticed error is present in the conclusions of the expert system.

PROBLEMS ENCOUNTERED DURING DEVELOPMENT

During development, a large variety of problems were encountered ranging from minor hardware problems to very serious problems that could impair future developments; in this section, we present the problems and the solutions found to these problems.

Operating system problems

The most serious problem we encountered was related to the operating system; in fact, the MS-DOS operating system and it's 640 kbytes memory restriction made it difficult to have the CAD environment call the verification expert system without exiting; at the present time, we have reached the limit of cohabitation of these two programs and that explains why the verification of missing element rule has yet to be done.

We don't feel that there is a good solution to this problem and workarounds are temporary solutions that don't last long; for future development or for a full scale implementation of the "integrated CAD substation design environment", we will recommend to switch to a more powerfull operating system such as OS/2 or XENIX.

Limited memory problems

This problem is minor and can be easily corrected: adding at least 2 Mbytes of expanded memory improves the cohabitation of both the CAD environment and the verification expert system. However, this solution is only appropriate for softwares that use the expanded memory (such as Autocad) and does not solve the restrictions of MS-DOS which are builtin limitations.

Poor performance from the system

During development and evaluation of the integrated tool, we reached a point where system performance was inadequate; an IBM-PC AT running Autocad (which is a large program) with calls to external programs (verification expert system) proved to be inappropriate. This being the case, we had to switch to a more powerful computer, the IBM PS2 model 80.

Expert system shell problems

The expert system shell, Rulemaster 2, due to the 640 kbytes memory restriction imposed by MS-DOS, was unable to support the full version of the verification expert system. Two solutions exist for this problem : whether a new version of Rulemaster using the expanded memory is released or the development of the expert system could

be done on another platform (a UNIX workstation for example) and the final product sent back in the MS-DOS environment.

Nevertheless, given the proper solutions, we feel that Rulemaster is a tool that still can be used to manage this expert system and improve it.

STATUS OF DEVELOPMENT AND FUTURE IMPROVEMENTS

The project has spawned over a period of 12 months and has required one full time developer for this period; given this, except for the verification of missing elements rule, all the other rules are coded in the verification expert system and minor adjustments have to be done (such as correcting rounding errors). Future upgrades should include the verification of missing elements and the validation of the graph of dependencies (within the step that builds the graph of dependencies) to check that the graph of dependencies is consistent.

As mentioned in the previous section, these improvements will be added to the verification expert system as soon as a solution to MS-DOS's memory restriction is reached.

CONCLUSION

Designing an "integrated CAD substation design environment" was a challenging task and, during development, we could not validate our concepts from past experience or other designer of such application's experience; although this product is still not "usable", considerable expertise has been gained in this domain and should help in the development of similar products. We believe that this product can be brought to a usable level with the collaboration of our client.

ACKNOWLEDGMENTS

The developers of this product would like to thank Mr. Claude Taliana and Henri Munger from Hydro-Quebec for their support during development; being our clients and considering the innovative nature of this project, their comprehensiveness and openness were key elements to the success of this project. Also, a special mention to Mr. Radu Manoliu, chairman of Hydro-Quebec's task force on expert system, who provided the developers with the appropriate computers and who managed this project.

MOAS II: An Intelligent On-Line Disturbance Analysis System

I. S. KIM* and M. MODARRES
Department of Chemical and Nuclear Engineering
University of Maryland
College Park, Maryland 20742, USA

ABSTRACT

A systematic model-based methodology for building a real-time expert system for on-line process disturbance management has been developed and presented elsewhere (1,2). In order to demonstrate its feasibility, the methodology has been applied to a typical main feedwater system used in pressurized water reactor (PWR) nuclear power plants. This paper describes the development of the models for the target process along with the implementation of the developed models in PICON (3,4) and the performance test of the resulting real-time expert system, MOAS II.

INTRODUCTION

A systematic methodology for building a real-time expert system for on-line process disturbance management--shortly an intelligent disturbance analysis system--has been presented in references (1) and (2). The methodology encompasses diverse functional aspects that are necessary for effective process disturbance management: intelligent process monitoring and alarming, sensor failure diagnosis, hardware (besides sensors) failure diagnosis, and corrective measure synthesis. Accomplishment of these functions is made possible through the integrated application of the various models: goal-tree success-tree (GTST), process monitor tree (PMT), sensor failure diagnosis tree (SFDT), and hardware failure diagnosis (HFD) modules.

GTST is a structure in which all the knowledge of the process plant and its operation relevant to the achievement of the top objective defined in the composite tree can be incorporated in a logical, hierarchical, and complete fashion. This structure is also used to identify process monitoring points, i.e., sensors that should be continuously or periodically monitored by the real-time expert system. However, it is not used on-line.

All the other models except the GTST, once implemented into the real-time expert system, will be used on-line for process disturbance management. PMT is a framework within which an intelligent process monitoring scheme can be incorporated; while, SFDT and HFD modules are the models in which sensor failure diagnosis and hardware failure diagnosis are performed.

In order to demonstrate the feasibility of the model-based methodology, it has been applied to a typical main feedwater system (FWS) of a nuclear power plant. This paper provides a description of: (i) the rational for the selection of the FWS as a target

*Currently with Brookhaven National Laboratory.

process in this work, (ii) the development of the models for the target process, (iii) the implementation of the developed models in PICON (3,4), and (iv) the performance test of the resulting expert system, MOAS II.

TARGET PROCESS

The FWS was selected as a target process for the application of the methodology for the following reasons:

1. In the operating history of the nuclear power plants, the malfunction of the FWS has been one of the most significant contributors to the plant outages (5). However, the partial loss of this system does not completely disable its heat removal capability, provided quick remedial actions are carried out. Therefore, there is a large incentive for developing an operator aid for process disturbance management in the FWS. The operator aid, if developed and employed in the control room, will also improve plant safety through the intervention of the fault propagation at its early stage and thereby the reduction of challenges to the safety systems.

2. Any process fault diagnosis method proposed should be applicable to control loops, because most complex process plants consist of a number of control loops and the difficulty in fault diagnosis mostly results from the complicated fault propagation structure in the control loops. The FWS employs a complex control mechanism; hence, the feasibility of the methodology can be tested and demonstrated by applying it to the FWS.

Figure 1 shows a schematic of the simplified version of the FWS used in this study. The primary objective of the FWS is to maintain proper water inventory in the steam generators during plant operations by supplying the required feedwater flow to each steam generator.

In order to achieve the objective, the FWS employs three kinds of controllers: flow controllers (FC511 and 512), differential pressure controllers (PDC511 and 512), and speed controllers (SC401 and 402). The flow controller (FC511) uses as its input the steam generator level (LT511), feedwater flow rate (FT511), and steam flow rate (FT611); whereas, the differential pressure controllers and the speed controllers implement a sort of auctioneering cascade control mechanism.

Most of the failures in the FWS have been especially due to the loss of a control circuit in the highly coupled and complex feedwater control system (6). For this reason, the emphasis in this work has been put on the feedwater control system.

MODELS FOR THE TARGET PROCESS

Figures 2 through 4 and Table 1 show some representative portions of the models developed for the target process; each of these will be briefly described in this section. See reference (1) for detailed descriptions of the models.

GTST for the FWS

The top objective defined for the development of a GTST is "Cycle Equivalent Availability Maximized," since the expert system is intended to be used to improve plant availability by early management of process disturbances. With this top objective, a goal tree was first constructed top-down following the development procedure (7,8). It may be noted here that an operator aid can also be developed for protecting loss of not only plant productivity but also safety. In this case, the

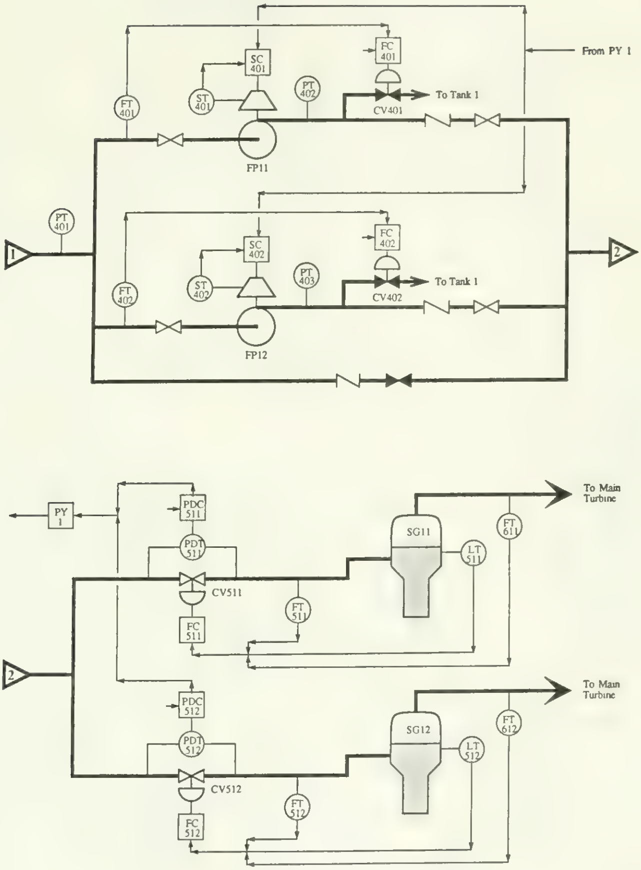


Figure 1. Schematic of the Simplified Feedwater System (Partial)

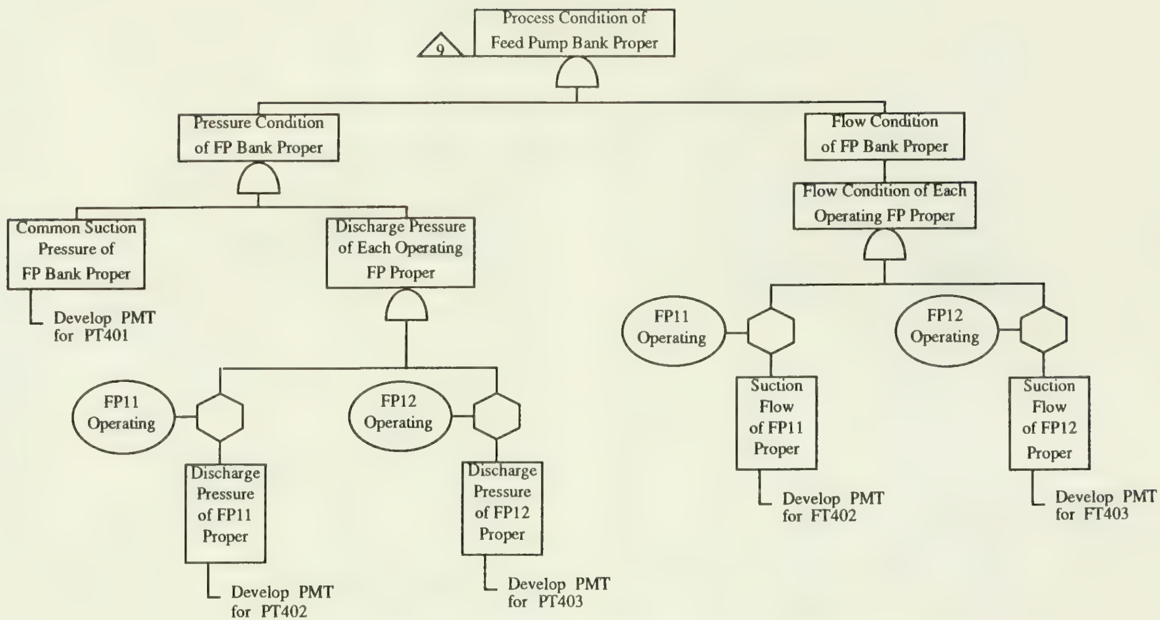


Figure 2. Goal-Tree Success-Tree Model for the FWS (Partial)

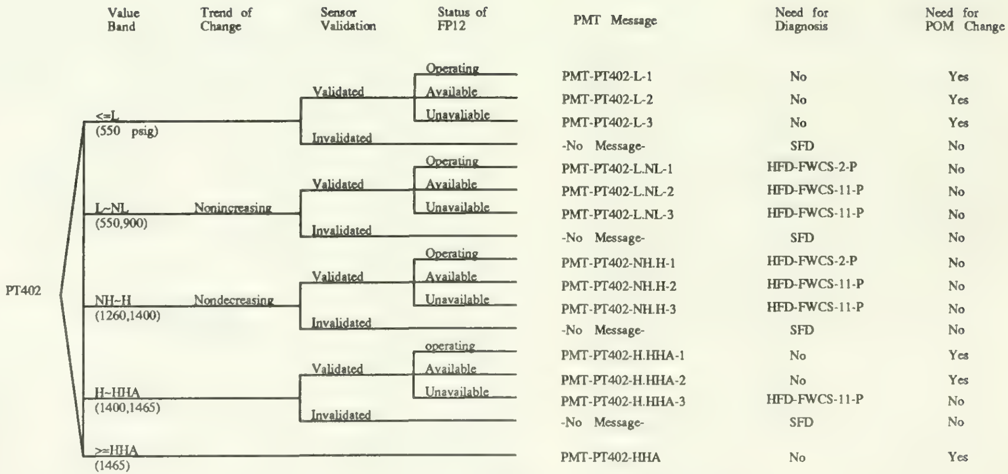


Figure 3. Process Monitor Tree for PT402

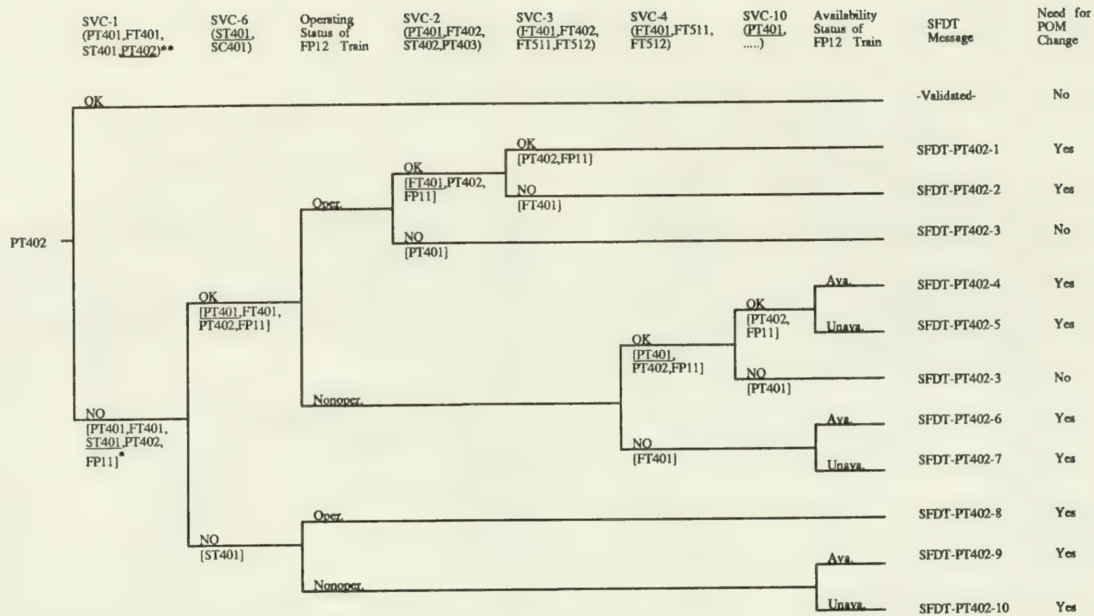


Figure 4. Sensor Failure Diagnosis Tree for PT402

Table 1

Hardware Failure Diagnosis Module

HFD Module	Failure Hypothesis	On-Line Verification	HFD Message Set	Need for POM Change
PT402.LOW-PT403.LOW	(1)* FM5	SC401, SC402, PDC511 in auto; PDC512 in manual; PDC511 setpoint low	[DG] PDC511 Setpoint Is Low [CM] Manually Increase FP11 & 12 Speeds	No
	(2) FM9	SC401, SC402, PDC512 in auto; PDC511 in manual; PDC512 setpoint low	[DG] PDC512 Setpoint Is Low [CM] Manually Increase FP11 & 12 Speeds	No
	(3) FM13	SC401, SC402, PDC511 in auto; PDC512 in manual; FC511 setpoint low	[DG] FC511 Setpoint Is Low [CM] Manually Open SG11 FW CV511	No
	(4) FM17	SC401, SC402, PDC512 in auto; PDC511 in manual; FC512 setpoint low	[DG] FC512 Setpoint Is Low [CM] Manually Open SG12 FW CV512	No
	(5) FM7	SC401, SC402, PDC511 in auto; PDC512 in manual; PDC511 output low; F-PDC511**	[DG] PDC511 Output Is Low [CM] Manually Increase FP11 & 12 Speeds	No
	(6) FM11	SC401, SC402, PDC512 in auto; PDC511 in manual; PDC512 output low; F-PDC512**	[DG] PDC512 Output Is Low [CM] Manually Increase FP11 & 12 Speeds	No
	(7) FM15	SC401, SC402, PDC511, FC511 in auto; PDC512 in manual; FC511 output low; F-FC511**	[DG] FC511 Output Is Low [CM] Manually Open SG11 FW CV511	No
	(8) FM19	SC401, SC402, PDC512, FC512 in auto; PDC511 in manual; FC512 output low; F-FC512**	[DG] FC512 Output Is Low [CM] Manually Open SG12 FW CV512	No
	(9) FM22	SC401, SC402, PDC511 in auto; PDC512 in manual; PDT511 high	[DG] PDT511 Is High [CM] Manually Increase FP11 & 12 Speeds	No
	(10) FM24	SC401, SC402, PDC512 in auto; PDC511 in manual; PDT512 high	[DG] PDT512 Is High [CM] Manually Increase FP11 & 12 Speeds	No

*These numbers indicate the predefined testing order of the hypotheses in the HFD module.
**These are deep process knowledge used to test failure hypotheses (see reference 11).

top objective will have to be expanded in such a way that it includes both safety and economy.

Among the goals identified in the goal tree as necessary for achieving the top objective, the goal, "Operability of the FWS Maximized," is directly relevant to the disturbance management of the target process. Hence, a success tree has been developed only for this goal.

Subtree 9 of the GTST in Figure 2 depicts a representative portion of the success tree. From the bottom nodes of the subtree which refer to process conditions, the sensors--PT401, PT402, PT403, FT402, and FT403--have been identified as process monitoring points. Similarly from the other part of the GTST, steam generator level sensors (LT511 and 512) have also been identified as process monitoring points. Feedwater flow sensors (FT511 and 512), steam flow sensors (FT611 and 612), and feedwater pump speed sensors (ST401 and 402), however, have not been identified as such sensors for which process monitor trees should be developed.

PMTs for the FWS

A PMT has been developed for each of the process monitoring points identified in the GTST. One of the PMTs--for the feedwater pump 11 discharge pressure sensor PT402--is shown in Figure 3. This PMT, once developed and implemented into the expert system, will continuously monitor the discharge pressure condition of feedwater pump 11 when it is in operation.

SFDTs for the FWS

Figure 4 depicts one of the SFDTs developed for the FWS, i.e., SFDT for PT402. Sensor failure is diagnosed by effective use of sensor validation criteria (SVCs) in the structure of the SFDT. The SVCs used as headings in the tree have been formulated from deep process knowledge such as conservation equations or pump performance curves. For example, the SVC-1 consisting of four process variables, i.e., values of PT401, FT401, ST401, and PT402, (see Figure 4) was formulated based on the performance curve of feedwater pump 11.

HFD Modules for the FWS

A typical of the HFD modules developed for the FWS is shown in Table 1. The HFD module of PT402.LOW-PT403.LOW will be activated on-line for hardware failure diagnosis in the case where a pressure disturbance has occurred in the FWS and it is such that the discharge pressures of both feedwater pumps are low. Upon the activation of the HFD module, the failure hypotheses will be tested by its associated on-line verification method following the predefined testing order, so as to determine the most likely cause that has brought about the disturbance pattern.

The development of the HFD modules was greatly facilitated by use of the simplified directed graph (SDG) depicted in Figure 5. The SDG, a simplified varient of the conventionally used digraph (9,10), models only the essential causalities of the FWS that are necessary and sufficient to extract the failure hypotheses for the process disturbances occurring in the system.

The FM nodes in the SDG represent failure modes, while the circled nodes denote the process variables. The signs in the SDG can be interpreted as in the digraph. For example, the positive sign between the process-variable node S401, i.e., feedwater pump 11 speed, and another process-variable node P402, i.e., feedwater pump 11

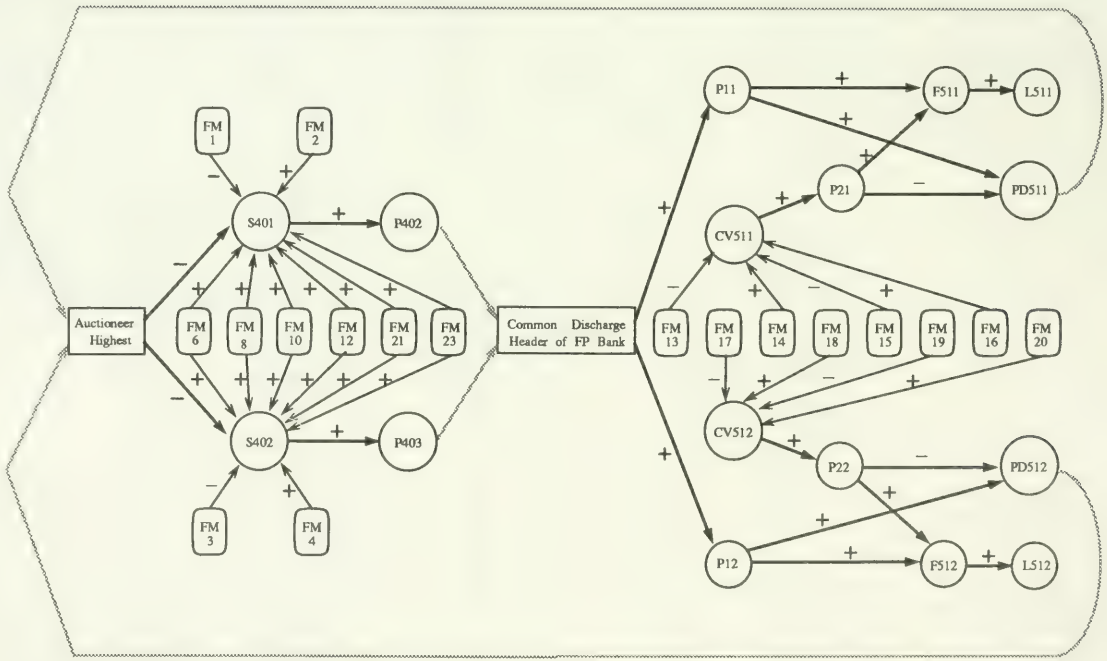


Figure 5. Simplified Directed Graph for the FWS

discharge pressure, indicates that the deviations of the process variables from the normal values occur in the same direction. Namely, it expresses that P402 will increase (decrease), if S401 increases (decreases). On the other hand, the negative sign between the failure-mode node FM17, i.e., flow controller FC512 setpoint being low, and the process-variable node CV512, i.e., opening of the feedwater regulating valve CV512, can be interpreted as such that the opening of the CV512 valve will be significantly and immediately influenced by the occurrence of FM17 and deviate from the normal values in a negative direction.

More details of the interpretation of the SDG, e.g., that of the special node "Auctioneer Highest," can be found in reference (11). It may be noted here that the SDG is not directly used as a model to develop the expert system, but used only as an aid to develop the HFD modules.

IMPLEMENTATION OF THE MODELS

One of the characteristics of the methodology is that implementation of the models and thereby construction of a real-time expert system can be done easily because of the well-developed knowledge representation structures and the model-based reasoning scheme incorporated in the models. The models developed for the FWS have been implemented in the real-time expert system development tool, PICON (Process Intelligent CONtrol) (3,4). The use of PICON in this work further simplified the development of the real-time expert system; as a result, the efforts could have been dedicated to the development and refinement of the methodology without getting involved with the smallest details of the programming in a language such as LISP or PROLOG (12).

This section briefly describes the PICON tool and the way by which the models have been implemented. See reference (11) for more details.

Real-Time Expert System Development Tool -- PICON

The following presents a brief description of PICON with an emphasis on the features which facilitate the development of a real-time expert system for process control applications:

1. The major knowledge representation structure of PICON is an IF-THEN rule frame consisting of an IF-THEN rule and its associated frame. The frame contains several slots such as timing or taxonomic slots. The timing slots, e.g., scan or alert intervals, are used to indicate when or how frequently the rule should be invoked, whereas the taxonomic slots, e.g., category, associated unit, or priority, are used for efficient access to rules. In addition to the IF-THEN rule frame, there are also other types of rule frames such as initial rule, whenever rule, or let statement. Initial rules perform actions when the knowledge base is first run; on the other hand, whenever rules carry out actions whenever a value varies more than some specified amount. Let statements are the only kind of rules which cannot perform actions; instead, these are used to define variables or conditions.

2. Every measured or inferred variable in PICON is "time-stamped," so that it is possible to know how old the information is. Furthermore, every variable has a "currency interval." Information about states older than the currency interval is not used by the inference engine. The need to know the state when its latest value is no longer current causes the state of the variable to be reevaluated in the current circumstances of the process condition.

3. The inference engine is deliberately designed to gather dynamic data from the data

supplier and allow the data to be applied to the rules in the knowledge base which require the data. For instance, if the inference engine cannot obtain up-to-date information from the data supplier upon the request by the knowledge base, it reminds itself to try again at some future time when its requisites may be available.

4. PICON provides a simulation environment in which the sensor values can be easily emulated in a dynamic fashion. The inference engine can get direct access to this simulation data through the data supplier. Therefore, knowledge base and inference control scheme of a real-time expert system can be easily tested in the simulation environment of PICON.

Implementation of PMTs

As a typical example of the implementation of PMTs, the first two conditions on the second branch under the first heading in the PMT for PT402 (Figure 3), i.e., L < PT402 ≤ NL and PT402 nonincreasing, were formulated in an IF-THEN rule and two definition statements as follows:

```
IF PT402.L.NL and not PT402.INCREASING                    (1)
THEN activate PT402.L.NL rules
[Associated Unit = FP11.TRAIN]

LET condition PT402.L.NL = (PT402 > 550 psig)              (2)
and (PT402 ≤ 900 psig)

LET condition PT402.INCREASING = (PT402 - PT402            (3)
as of 9 seconds ago) / PT402 > .01
```

Rule (1), one of the front-end rules of the PT402 PMT, will be activated by the following primary rule, as long as the antecedent of the primary rule tests true:

```
IF SSF is GREEN and FP11.TRAIN is OPERATING                (4)
THEN activate rules with associated unit =
FP11.TRAIN
[Scan Interval = 5 seconds]
```

Therefore, the discharge pressure condition of the feedwater pump 11 will be monitored every five seconds by the front-end rules of the PT402 PMT such as rule (1) when the pump is in service. The SSF, i.e., system status flag, in the antecedent of rule (4) is one among the three kinds of flags used to control the multiple inference processes; the other two kinds are sensor failure diagnosis flags and hardware failure diagnosis flags (11).

Implementation of SFDTs

The implementation of the SFDT for PT402 (Figure 4) resulted in thirteen IF-THEN-ELSE rules and several LET statements. The OK branch under the SVC-2 heading can be programmed in PICON as follows:

IF abs (PT403 - PT401 - 0.433 * (-0.00004 * (5)
$FT402^2$ + 0.49 * FT402 + 195 + 1.5667 *
(ST402 - 4800))) < FP.CURVE.TOL
THEN activate PT402-SFD-4 rules
ELSE activate rules with priority = 101
[Categories = PT402-SFD-3]

The expression in the IF part of the rule represents the SVC-2 based on the feedwater pump 12 performance curve. Depending on the test result of the SVC-2, the inference engine will activate by forward chaining either PT402-SFD-4 rule or rule with priority = 101; these two rules are the formulation of the subsequent two branches, i.e., the OK and NO branchs under the SVC-3 heading (see Figure 4).

Implementation of HFD Modules

In the case of the PT402.LOW-PT403.LOW module (Table 1) discussed above, the second failure hypothesis with its associated on-line verification method and HFD message set can be implemented as follows:

```
IF SC401.OPER.MODE is AUTO and                                    (6)
SC402.OPER.MODE is AUTO and
PDC511.OPER.MODE is MANUAL and
PDC512.OPER.MODE is AUTO and
PDC512.SETPOINT.LOW and SSF is GREEN
THEN send "[DG] PDC512 SETPOINT IS LOW
[CM] MANUALLY INCREASE FP11 & 12 SPEEDS"
to operator and advance step variable SSF
to BLACK
ELSE activate PT402.LOW-PT403.LOW-3 rules
[Categories = PT402.LOW-PT403.LOW-2]

LET condition PDC512.SETPOINT.LOW =                               (7)
(PDC512.SETPOINT ≤ 100)
[Currency Interval = Derived]
```

Should all the conditions of the antecedent of rule (6) test true, the diagnosis process in the HFD module will be terminated after presenting the message set in the THEN part of the rule to the operator and advancing the step variable SSF to black. Otherwise, the third failure hypothesis in the HFD module will then be tested by the rule with a category of PT402.LOW-PT403.LOW-3. The LET statement is invoked from rule (6) by backward chaining when the PDC512.SETPOINT.LOW condition is encountered during the test of the antecedent of the rule.

PERFORMANCE TEST OF MOAS II

A real-time expert system for on-line process disturbance management in the FWS, MOAS II, has been developed on the LMI LAMBDA Machine (3,4) by implementing the models for the FWS as discussed above and using the flag concept (11). MOAS II is an advanced follow-up of our CFWAVA (8), a pilot expert system written in PROLOG artificial intelligence language, and MOAS (13), a prototype real-time expert system and precursor of MOAS II developed also in PICON.

Figure 6 shows the development process and structure of MOAS II. The domain knowledge needed for the on-line process disturbance management is captured in the various models such as GTST, PMTs, SFDTs, and HFD modules. The implementation of the models in PICON constitutes the knowledge base of MOAS II.

Another constituent of MOAS II is the inference engine; for the demonstrative expert system the real-time inference engine of PICON was used. However, flags were embedded in the knowledge base in order to control the multiple inference processes performed by the inference engine. The real-time inference engine requests the data supplier to provide the on-line sensor data it needs, and then applies the sensor data acquired from the data supplier to the relevant rules in the knowledge base.

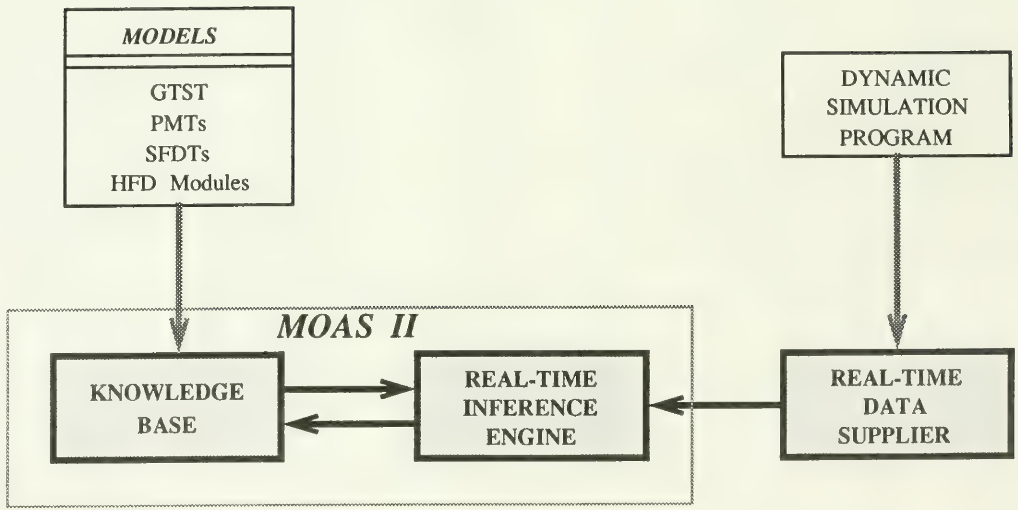


Figure 6. Development Process and Structure of MOAS II

A dynamic simulation program for the operation of the FWS--normal and abnormal--was developed in BASIC to prepare the data supplier. The simulation program was run on an IBM/PC computer to get the dynamic system behavior in terms of process variable or parameter values such as sensor values or controller outputs. The obtained system behavior was then emulated by use of the simulation facility of PICON, so that the inference engine of PICON can get access to the simulation data.

The performance of MOAS II was tested in this simulated process environment against a variety of transient scenarios resulting from failures of sensors, pumps, or control elements. Figure 7 shows the transient behavior of the FWS following the occurrence of a high-biased failure of the flow controller FC511 output at t = 30 sec, in terms of only four key process variables: discharge pressures of feedwater pumps 11 and 12 (PT402 and PT403), and water levels of steam generators 11 and 12 (LT511 and LT512). It is shown in the figure that if the process anomaly is not rectified in a timely manner, then the main turbine and subsequently the whole plant will be tripped upon the high water level in the steam generator 11.

The performance of MOAS II was tested in the transient process environment represented in Figure 7. The real-time messages presented by MOAS II are shown in Table 2. At 93 sec, MOAS II prealarms the operator of the moderately high steam generator 11 level with the indication of the steam generator 11 level at that time point, i.e., 59.9 cm (23.6 inches) above the normal level. In addition, it also prealarms him of the turbine/plant trip that may follow if the disturbance is not corrected in due time.

About six seconds after the presentation of the first message set, MOAS II provides the operator with the second message set which contains the diagnosis result of FC511 output failing high and an appropriate control advice, i.e., manual closing of the steam generator 11 feedwater control valve CV511.

The Model column of the table is not actually presented by MOAS II; however, it is included in the table to indicate the origin of the message sets. The prealarming message set originates from the PMT for LT511; whereas, the diagnosis and corrective measure message set is given from the HFD module of LT511.HIGH-LT512.NORM in which the hardware failure diagnosis was performed.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Engineering Research Centers Program NSFD CDR 88003012, and by Baltimore Gas and Electric Company.

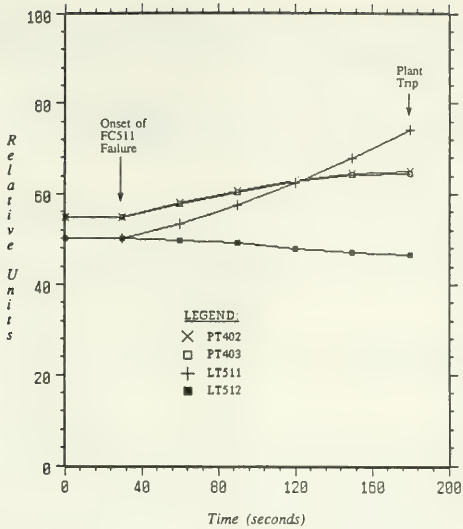


Figure 7. Simulation of the Transient System Behavior Resulting from a High-Biased Failure of Flow Controller FC511 Output

Table 2

Real-Time MOAS II Message Sets and Their Origins

Time (sec)	MOAS II Message	Model
93	[PA] SG11 Level Mod High (LT511=23.6) Turbine/Plant Trip at 50 inches	PMT (LT511)
99	[DG] FC511 Output Is High [CM] Manually Close SG11 FW CV511	HFD (LT511.HIGH-LT512.NORM)

REFERENCES

1. I. S. Kim, M. Modarres and R. N. Hunt. "A Model-Based Approach to On-Line Process Disturbance Management: The Models." To Appear in Reliability Engineering and System Safety.

2. I. S. Kim and M. Modarres. "Toward Building Real-Time Expert Systems for On-Line Process Disturbance Management Using Model-Based Reasoning." Proc. Seventh Power Plant Dynamics, Control and Testing Symposium, Knoxville, TN, May 1989.

3. R. Moore et al. "Expert Systems in Real-Time Environments." Proc. International Conference on Computer Applications for Nuclear Power Plant Operation and Control, Tri-Cities (Pasco), WA, September 1985.

4. R. Moore et al. "A Real-Time Expert System for Process Control." Proc. IEEE First Conference on Artificial Intelligence Applications, Denver, CO, 1984.

5. C. H. Meijer and B. Frogner. On-Line Power Plant Alarm and Disturbance Analysis System. EPRI Report, Electric Power Research Institute, Palo Alto, CA, NP-1379, April 1980.

6. C. W. Solbrig. "Significant Event Experience Base." Nuclear Engineering and Design, Vol. 83, 1984, pp 165-187.

7. R. N. Hunt and M. Modarres. "Integrated Economic Risk Management in a Nuclear Power Plant." Annual Meeting of the Society for Risk Analysis, Knoxville, TN, October 1984.

8. I. S. Kim and M. Modarres. "Application of Goal Tree-Success Tree Model as the Knowledge-Base of Operator Advisory Systems." Nuclear Engineering and Design, Vol. 104, 1987, pp 67-81.

9. S. A. Lapp and G. J. Powers. "Computer-Aided Synthesis of Fault-Trees." IEEE Transactions on Reliability, April 1977, pp 2-13.

10. E. J. Henley and H. Kumamoto. Designing for Reliability and Safety Control. Englewood Cliffs, NJ: Prentice Hall, Inc., 1985.

11. I. S. Kim, M. Modarres and R. N. Hunt. "A Model-Based Approach to On-Line Process Disturbance Management: The Application." Presented for Publication in Reliability Engineering and System Safety.

12. D. Hu. Programmer's Reference Guide to Expert Systems. Indiana: Howard W. Sams & Co., 1987.

13. I. S. Kim and M. Modarres. "MOAS: A Real-Time Operator Advisory System." Proc. International Conference on Artificial Intelligence and Other Innovative Computer Applications in the Nuclear Industry, Snowbird, Utah, August 1987.

TRESCL Expert System Software

CHARLES P. HORNE
CPH, Inc.

DAVID G. CAIN and BILL K. H. SUN
Electric Power Research Institute
3412 Hillview Avenue
Palo Alto, California 94304, USA

ABSTRACT

This paper presents the results of a research project (RP 2395-6) sponsored by the Electric Power Research Institute (EPRI) to develop full featured, high performance, customizable, hardware independent expert system software in the C programming language.

The program developed in this project was named TRESCL (Tool for Recasting Expert System to C Language), connoting a metaphorical trestle bridge joining AI technology to practical applications. TRESCL was designed to emulate EPRI's SMART (Small Artificial Reasoning Toolkit) expert system software.

TRESCL combines features of C and Lisp, and is designed for applications that have demanding time constraints, or that need to be tightly integrated with other software or hardware, or that need to run on computers with otherwise inadequate expert system support, or that need customized inferencing algorithms.

1. INTRODUCTION

Expert systems have been gaining acceptance as a technique for solving problems in the electric utility industry. However, some impediments to further use of expert systems persist. These impediments include specialized hardware requirements, slow performance, incompatibility with existing software, and inaccessibility to the inference algorithms. The software described in this paper addresses these issues and provides an alternative approach for utilities to access expert system technology.

TRESCL's capabilities were modeled from SMART (1), which is written in Lisp and incorporates many features of larger systems, such as KEE (3). SMART does more than execute rules: it provides an object-oriented modeling environment. A SMART

object may represent a physical plant component, or a process, or a reasoner, which surveys other objects, applies rules, forms opinions, and induces actions. Each object has a set of attributes, called slots, which may contain values or procedures (called methods). Objects communicate with each other via messages. Methods respond to messages to give an object behavior. Values and methods may be inherited from any of several parents. Demons are special methods that are automatically triggered when some predefined event occurs.

SMART provides a user interface to guide the construction of models. When the developer gives an object rules, SMART automatically creates certain slots in the object to hold the rules, the forward or backward chaining methods, and the beliefs generated by the rules.

Like SMART, TRESCL supports rules, objects, methods, demons, and side effects. TRESCL compiles rules, objects, and object properties into a C data structure optimized for high speed inferencing. TRESCL contains a small Lisp interpreter to load SMART knowledge bases and evaluate SMART expressions.

Some of the properties of the TRESCL expert system toolkit are :

- full featured : TRESCL's features include object-oriented modeling, multiple inheritance, forward and backward rule inferencing, rule side effects, methods, hypothetical reasoning, non-monotonic reasoning, demons, variable binding, a variety of inference controls, and Lisp evaluations. TRESCL has rule and object browsers and can formulate explanations.

- high performance : TRESCL's inference algorithm is optimized for speed. Rules are compiled into a network to eliminate most run-time searching .

- portable : TRESCL is written in ANSI C, so that any computer with a C compiler can run TRESCL. TRESCL will run on a Cray or an embedded instrumentation microprocessor.

- customizable : TRESCL can be customized with application specific objects, rule clauses, methods, side effects, demons, or by modifying inference options, or by modifying TRESCL itself.

- compatible : Because it is written in C, TRESCL is compatible with most software used for conventional applications. Functions written in C, FORTRAN, or Pascal can be used as rule clauses, methods, side effects, or demons. These functions can provide fast access to existing data base, computation, communication, graphics, or data acquisition software.

The principal input to TRESCL is the knowledge base created by SMART, as illustrated in figure 1. The other inputs to TRESCL are the interactive user commands, and other application specific data.

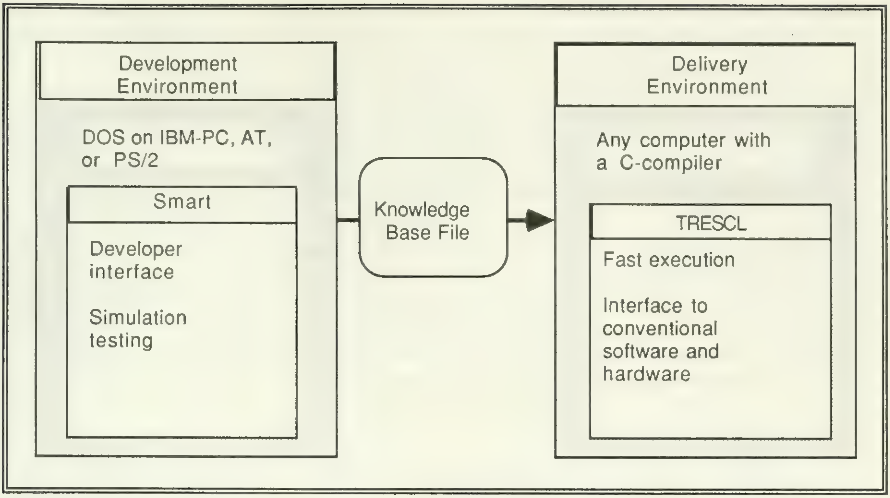


Figure 1. TRESCL Application Development

1.1 Background

This study began in July, 1986 with a prototype of EPRI/TaiPower's Emergency Operating Procedures (EOP) tracking system (3, 4). The EOP prototype system was an 'application generator', that translated EOP rules into C source code file that was compiled and linked to produce an executable program. The EOP system had only a small portion of SMART's capabilities , but its success led to the more ambitious TRESCL program to emulate all of SMART's functionality. Development of TRESCL ended in November, 1987.

1.2 Implementation

The strategy of the TRESCL project was to write a small Lisp interpreter in C, combining features of C with features of Lisp. Lisp knowledge bases from SMART are read into a Lisp in-memory representation which is then complied to use C data structures and functions that optimize inferencing. TRESCL's internal representation of objects and rules are more C-like than Lisp-like, but the Lisp representation is preserved to support Lisp evaluations and interpretative invocations of C functions.

As illustrated in figure 2, TRESCL has three principal components : an object system, a rule system, and a Lisp interpreter. TRESCL also includes a minimal, line-oriented user interface that allows the user to send messages and inspect rules, terms (rule clauses), and objects. It was assumed that for each application, the user interface would be replaced by an application driver.

The Lisp system within TRESCL is minimal, consisting of only those primitives (about 20) necessary for SMART knowledge bases. As each symbol is read, it is looked up to see if it refers to an previously encountered symbol. The Lisp reader passes a thread through each instance of a symbol. Each symbol instance is counted, so that when future dereferences occur (as in rebinding), the symbol may be removed from memory if it is unreferenced. This process is known as garbage collection. After the knowledge base file is read, TRESCL traverses and modifies the Lisp representation to link function references and to restructure objects, rules, and rule clauses. When a LAMBDA or DEFUN is encountered, a binding structure is created to enable deferred argument binding. When the knowledge base compilation is complete, TRESCL is ready for run-time operation.

Custom functions written in C or FORTRAN may be added easily and accessed via the Lisp interface. Arguments are passed from the interpreter to functions as a Lisp expression which is decomposed in the function preamble.

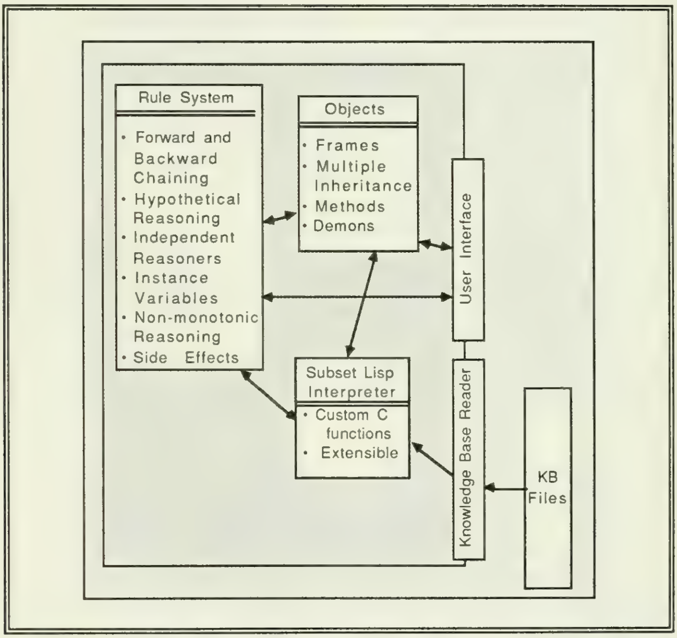


Figure 2. Components of TRESCL

Rules in SMART/TRESCL have the standard form :

(IF (*premise 1*) ... (*premise N*))
(THEN (*conclusion*) (*side effect*)))

where *premise* , *conclusion* and *side effect* are Lisp expressions. A rule term (*premise* or *conclusion*) may be expressed as any of the following forms :

- a simple clause, such as SRV PUMP IS ON
- a relational, such as SRV PUMP PRESSURE > 50
- a function, such as ANY (SRV PUMP IS ON)
- an arbitrary Lisp expression in the form of a call to the Lisp evaluator, such as LISP(EQ(LAST(GET(SRV PUMP)) 'ON))

Rules are powerful because their intuitive meanings are simple, but their precise meaning can be quite complex in aggregate and under various assumptions about rule completeness, negation, failure, retraction, an so forth (5, 6). TRESCL provides a small set of options to control the reasoning process.

Each rule forms a node in the network that is connected to other nodes by premise and conclusion terms represented as directed arcs. Each node has private data used to determine whether the rule can propagate. The memory contains vectors representing the current status and the status required for propagation. The vector representation is an abstraction of the state that allows rapid comparisons to be made during the inference process. Each term has pointers to all rules referencing the term, and each rule has pointers to all terms it references.

During inferencing, truth states are propagated through the network in a computational wave with two phases : rule propagation and term evaluation. Forward and backward chaining use the same propagation routines, but have different initialization procedures and different success vectors.

The forward chaining inference begins by evaluating the truth of an initial set of terms whose values are known. The TRESCL forward chainer uses three logic states : TRUE, FALSE, and UNKNOWN. TRESCL detects and reports any conflicts in an audit file. One type of conflict detected occurs when a truth state changes from TRUE to FALSE or conversely. When the truth states of the premise terms of a rule match the patterns of TRUE and FALSE required for success of the rule, then the appropriate truth states of the conclusion terms are assigned. As an option, the slot values corresponding to conclusion terms may also by set if the rule succeeds. SMART and TRESCL allow multiple terms ANDed together in premises, and any term may be negated. In cases where more than one premise may lead to the same conclusion (the equivalent of OR), several distinct rules must be written.

The backward chaining inference uses two additional truth states : H_TRUE and H_FALSE. The backward chaining inference begins by setting the truth states of a set of hypotheses (declared by the human expert in the knowledge base file) to H_TRUE (or H_FALSE). The backward chainer then tries to confirm those hypotheses from other facts. To do so, it first links the hypothetical terms to rules which have conclusions that would confirm the hypotheses. If it finds such a rule, it marks the rule and its premises as H_TRUE (or H_FALSE), then repeats the process. Eventually, a set of terms which are not the conclusions of any rule are reached. These base terms are then determined from associated slot values, or from external data sources. When these terms are determined, all traces of hypothetical truth states are erased and the

forward chainer is invoked. If the truth states of the base terms imply the original hypotheses, then the hypotheses are proven.

Side effect functions are procedures that are invoked if a rule succeeds. A side effect may, for example, set hypotheses and induce forward and backward chainers, or they may send messages to other objects.

TRESCL may be used without modification as a stand-alone shell for some applications, but its greatest value is expected to be for applications requiring customization. TRESCL can be interfaced directly to computational, communication, data base, graphics, data acquisition, or other application specific software. Most commercial expert system shells provide only program-to-program communication via file interfaces, which are much slower.

1.3 Comparative Performance

Benchmark tests shown in figure 3 indicate that TRESCL is orders of magnitude faster than SMART for large numbers of rules. The times shown do not include the time to load and (in the case of TRESCL) compile the knowledge bases. The rule systems used for benchmarking were generated by a rule generation program. As with any benchmark, there are biases. The form of the benchmark was a string of rules that were deep and narrow - each rule depended on the conclusion of one other rule. The type of rule dependency burdens SMART's algorithm more than TRESCL's. A second bias is that SMART was benchmarked in interpretative, not compiled mode. A compiled version of SMART would be expected to be around 30 times faster. SMART's algorithm could also be modified to be more efficient for this type of benchmark, but no SMART applications have been limited by speed, since most SMART applications involve fewer than 50 rules.

	Time (sec)		
# rules	SMART	TRESCL	ratio
40	17	.33	51
140	411	.44	934
290	3318	.71	4673

Times in seconds

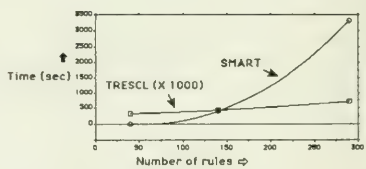


Figure 3. Benchmark Results

2. EXAMPLE APPLICATION : PWR BREAK IDENTIFIER

The PWR break identifier application is a simple example of how TRESCL could be used. The application was first developed with KEE on a Lisp machine as a training exercise, then converted to SMART, then to TRESCL.

The relevant components of a PWR reactor are shown in figure 4. The major conceptual objects identified by the analyst are shown in figure 5. Some of these objects are tangible, such as "PRESSURIZER", while others, like "TEST" are intangible. Each object has a number of attributes, represented as slots, which characterize the features of the object. Some slots hold data values, while other slots hold methods. The rules are shown in figure 6 in text form, and in figure 7 in an equivalent graphical form. In this case, all the rules are within the single object TEST. The object SMART contains methods that control the program.

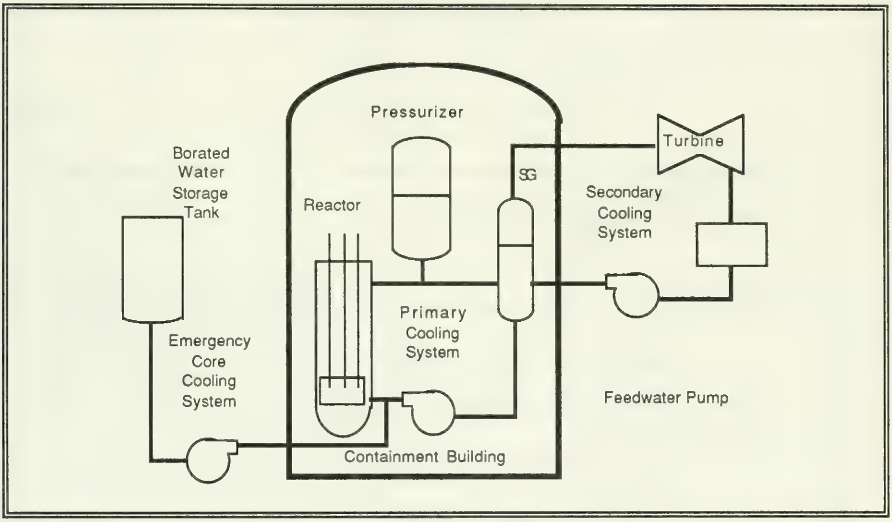


Figure 4. PWR components

In figure 7, terms are represented as nodes and rules are represented as joins of arcs. For example, the node labeled by the term "Pressurizer Pressure and Level Decreasing" is a premise to three rules, R1, R2, and R3, which are related to three conclusion terms. An N adjacent to an arc represents negation. For example, R3 has the term NOT (STEAM-GENERATOR PRESSURE IS ABNORMALLY LOW) as a premise. Several rules leading to the same conclusion can be depicted as unjoined arcs, as in the case of PRIMARY SYSTEM BOUNDARY IS BROKEN, which is a conclusion of both R3 and

R9. TRESCL encodes the nodes, arcs and success conditions much in the same way as described above.

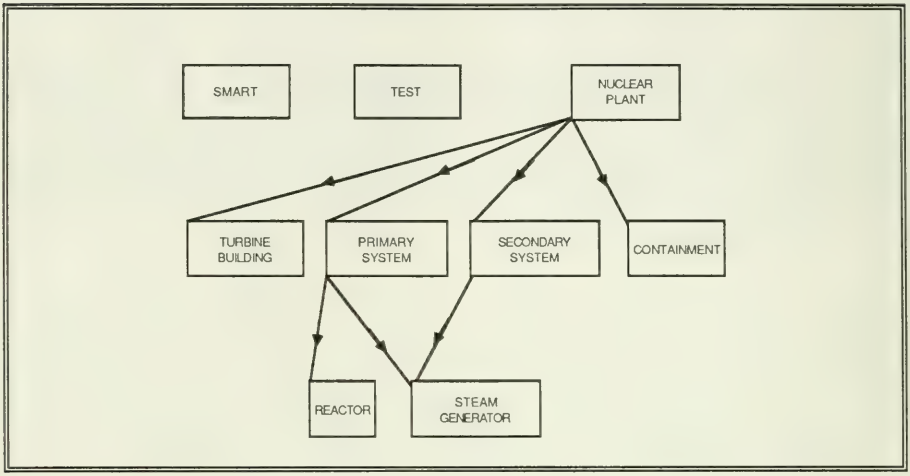


Figure 5. Break Identifier Object Structure

```
(RULE TEST 1
  (IF (NOT (PRESSURIZER LEVEL AND PRESSURE ARE DECREASING)))
  (THEN (SYSTEM-BOUNDARY IS NOT-BROKEN)))

(RULE TEST 2
  (IF (PRESSURIZER LEVEL AND PRESSURE ARE DECREASING)
     (STEAM-GENERATOR PRESSURE IS ABNORMALLY LOW))
  (THEN (SECONDARY SYSTEM BOUNDARY IS BROKEN)))

(RULE TEST 3
  (IF (PRESSURIZER LEVEL AND PRESSURE ARE DECREASING)
     (NOT (STEAM-GENERATOR PRESSURE IS ABNORMALLY LOW)))
  (THEN (PRIMARY SYSTEM BOUNDARY IS BROKEN)))

(RULE TEST 4
  (IF (SECONDARY SYSTEM BOUNDARY IS BROKEN)
     (CONTAINMENT PRESSURE IS INCREASING))
  (THEN (STEAMLINE IS BROKEN INSIDE CONTAINMENT)))

(RULE TEST 5
  (IF (SECONDARY SYSTEM BOUNDARY IS BROKEN)
     (NOT (CONTAINMENT PRESSURE IS INCREASING)))
  (THEN (STEAMLINE IS BROKEN OUTSIDE CONTAINMENT)))

(RULE TEST 6
  (IF (PRIMARY SYSTEM BOUNDARY IS BROKEN)
     (CONTAINMENT PRESSURE IS INCREASING))
  (THEN (LOSS OF COOLANT ACCIDENT IS INSIDE CONTAINMENT)))

(RULE TEST 7
  (IF (PRIMARY SYSTEM BOUNDARY IS BROKEN)
     (NOT (CONTAINMENT PRESSURE IS INCREASING))
     (TURBINE-BUILDING ACTIVITY IS NOTICABLY HIGH))
  (THEN (STEAM-GENERATOR TUBE IS RUPTURED)))

(RULE TEST 8
  (IF (PRIMARY SYSTEM BOUNDARY IS BROKEN)
     (NOT (CONTAINMENT PRESSURE IS INCREASING))
     (NOT (TURBINE-BUILDING ACTIVITY IS NOTICABLY HIGH)))
  (THEN (SMALL-BREAK LOSS OF COOLANT ACCIDENT IS OUTSIDE CONTAINMENT)))

(RULE TEST 9
  (IF (REACTOR WATER LEVEL IS VERY LOW))
  (THEN (PRIMARY SYSTEM BOUNDARY IS BROKEN)))
```

Figure 6. Text Representation of Break Identifier Rules

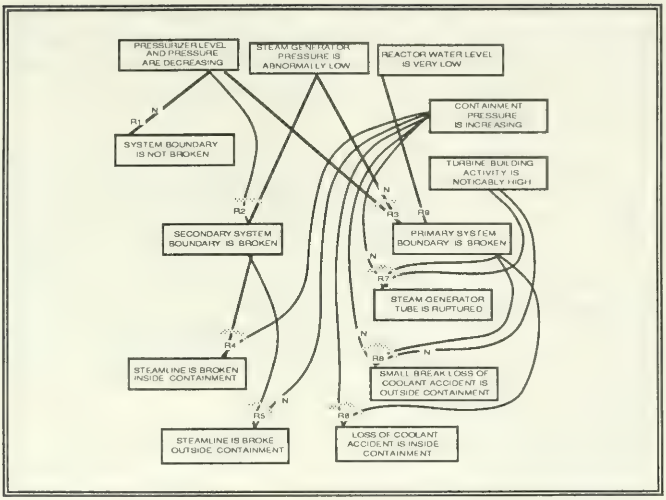


Figure 7. Graphical Representation of Break Identifier Rules

TRESCL is executed with the command TRESCL <kb> , where <kb> is name of the knowledge base file. The user first sees the program title as

```
Translate Expert System to C Language
Copyright EPRI 1987
EPRI Project Manager David Cain
Developed by Charles P. Horne, Inc
Mar, 1987
```

then the top level menu appears as :

```
Select Object Browser operation
1. Send message
2. Reset KB
3. Show objects
4. Show slot values
5. Show all slots with demons
6. Show all methods
7. Set slot values
8. Browse rule network
9. Other stuff
Q. quit
```

The first choice "1. Send message" activates the knowledge base. Messages sent to a method activate that method. Many SMART knowledge bases, including RX.KB, have a default method T in the SMART object as a convenient way to start an interactive session. If this option is chosen (by typing 1), then the user is asked for the object to which to send the message :

```
You chose 1, '  1. send message'

 select frames
    1 : all frames shown
    2 : 'SMART'
    3 : 'REACTOR'
    4 : 'STEAM-GENERATOR'
    5 : 'TEST'
    6 : 'TURBINE-BUILDING'
    7 : 'CONTAINMENT'
    8 : 'SECONDARY-SYSTEM'
    9 : 'PRIMARY-SYSTEM'
   10 : 'NUCLEAR-PLANT'
type numbers of frames (type CR to leave unchanged) : 2
```

Here, the user types 2 to select the SMART object. Next the user is asked to select a method within the selected object. In this example, the SMART object has two methods, SLIST and T.

```
slots for frame SMART :  select slots
   1 = 'SLIST'
   2 = 'T'
type numbers of slots 2
```

Here the user types 2 to select T, the default SMART trigger method. The method associated with this slot is defined in the knowledge base as :

```
(METHOD: TRIGGER)
 (MESSAGE TEST TXT)
 (SMENU TEST)
```

The clauses (MESSAGE TEST TXT) and (SMENU TEST) are Lisp clauses that constitute the method. MESSAGE is a function that displays the content of the slot TXT in the target object, TEST. SMENU presents a menu of all methods of the target object, TEST. In SMART the functions MESSAGE and SMENU are written in Lisp, and in TRESCL, they are written in C. TRESCL methods can generally be written in either Lisp or C. When they are written in Lisp, they are considered part of the knowledge base. For this example, the MESSAGE clause causes the following text to be displayed to the user:

```
message @Thu Aug  6 1987 13:58:02.03
===================================================

   *******WELCOME TO REACTORS EXPERT*******

THIS EXPERT SYSTEM DEMONSTRATES BACKWARD CHAINING
BY PATTERN MATCHING.  FOUR HYPOTHESES ARE TESTED:
LOSS OF COOLANT ACCIDENT INSIDE CONTAINMENT.
SMALL-BREAK LOSS OF COOLANT ACCIDENT OUTSIDE
CONTAINMENT.  STEAMLINE BREAK OUTSIDE CONTAINMENT.
STEAMLINE BREAK INSIDE CONTAINMENT.  BACKWARD
CHAINING IS INITIATED USING MENU OPTION B.  OPTION
F CAN BE USED TO SHOW FACTS WHICH ARE CONCLUDED BY
THE INFERENCING PROCESS.  OPTION G PRODUCES
EXPLANATION GRAPHS.  OPTION R RESETS THE KNOWLEDGE
BASE.
```

The Lisp clause (SMENU TEST), presents to the user a menu of available methods in the object TEST. Thus, the user next sees the menu :

```
Select a method for unit TEST
 1 RETURN-TO-SMART
 2 SHOW-FACTS
 3 RESET
 4 BKCHAIN
 5 RULEGRAPH
```

Some options, such as RULEGRAPH, are unimplemented. The option, "4 BKCHAIN" is the one that is most interesting - it invokes the backward chainer. The definition of this method is :

```
(METHOD: BKCHAIN)
(BKCN TEST)
```

The method item (BKCN TEST) invokes the backward chainer for rules in the TEST object. The backward chaining process begins with a hypothesis and seeks to determine facts that support he hypothesis. The backward chainer assumes that the hypotheses will be found in a slot called HYPOTHESES in the target object. The slot HYPOTHESES in the TEST object contains the hypotheses :

(LOSS OF COOLANT ACCIDENT IS INSIDE CONTAINMENT)
(SMALL-BREAK LOSS OF COOLANT ACCIDENT IS OUTSIDE CONTAINMENT)
(STEAMLINE IS BROKEN OUTSIDE CONTAINMENT)
(STEAMLINE IS BROKEN INSIDE CONTAINMENT)

Each of these will be processed in sequence. The activity of the backward chainer looks as follows :

```
Hypothesis = LOSS OF COOLANT ACCIDENT IS INSIDE CONTAINMENT
SetHypotheses : Hypothetical terms  :
  T : LOSS OF COOLANT ACCIDENT IS INSIDE CONTAINMENT
  T term = CONTAINMENT PRESSURE IS INCREASING
 Is the above term true, false, or unknown (TYPE t, f, u): t
  T term = REACTOR WATER LEVEL IS VERY LOW
 Is the above term true, false, or unknown (TYPE t, f, u): t
Hypothesis LOSS OF COOLANT ACCIDENT IS INSIDE CONTAINMENT
      is proven true
All hypotheses were proven

Select option
1. show why
2. reset KB
3. repeat inference
4. Browse rules
5. Change controls
6. proceed to next
```

In the above dialog, the user was asked whether two terms were true, false, or unknown. If the terms referenced existing slots, and the slot values had not been NIL, then the user would not have been asked for the values. In a real application, the resolution of truth could be determined by querying data bases, instruments, or computational servers.

At this point, the first hypothesis has been processed and proven. The user is being asked whether to go to the next hypothesis "6. proceed to next" or to do something else. The option "1. show why" is one of the more interesting ones. If selected at this time, this option would yield :

```
explain t term LOSS OF COOLANT ACCIDENT IS INSIDE CONTAINMENT  because of
rule 'TEST 6 ' :
  IF   ((PRIMARY SYSTEM BOUNDARY IS BROKEN)   and
       (CONTAINMENT PRESSURE IS INCREASING))
  THEN ((LOSS OF COOLANT ACCIDENT IS INSIDE CONTAINMENT))
 Type any character to continue
explain t term PRIMARY SYSTEM BOUNDARY IS BROKEN
 because of rule 'TEST 9 ' :
   IF   ((REACTOR WATER LEVEL IS VERY LOW))
    THEN ((PRIMARY SYSTEM BOUNDARY IS BROKEN))
```

3. SUMMARY

TRESCL is an innovative approach to providing expert system technology in a form which combines the best features of C with the best features of Lisp. In retrospect, the difficulty of verification was underestimated. The combinatorial complexity of even simple rule systems, amplified by several inference control options, yields a system that is difficult to verify in the traditional manner. The difficulty in determining the correct final state for a given initial condition may be inherent to intelligent systems.

The basic technical goals of this project were met. The types of applications that might be appropriate for TRESCL include :

- Applications requiring high speed inferencing with a large number of rules or high speed data acquisition, such as alarming.
- Supplementing existing numerical codes with rules.
- Adding rules to embedded applications, e.g. intelligent instrumentation or communications.
- Hybrid applications where the standard pattern deviates from the IF/THEN model of rule systems, or where access to the inference algorithm is necessary, e.g., adding learning weights.
- Porting expert systems to computers not supported by existing shells, or supporting hardware independent expert system capabilities.

References

(1) D. Cain, SMART : Small Artificial Reasoning Tool, EPRI 1985

(2) IntelliCorp, KEE : Software Development System User's Manual, Document number 3.0-U-1, IntelliCorp, 1986

(3) C. Horne, An Emergency Operating Procedures Logic Translator and Evaluator, EPRI Draft Report, June, 1986

(4) W. Petrick et. al., Emergency Procedures Tracking System for Nuclear Power Plants, ANS Topical Meeting "Artificial INtelligence and other Innovative Computer Applications in the Nuclear Industry", Snowbird Utah, September, 1987 and 1987 AI workshop.

(5) M. Genesereth & N. Nilsson, Logical Foundataions of Artificial Intelligence, Morgan Kaufmann Publishers, Los Altos, Ca., 1987

(6) L. Wos, R. Overbeek, E. Rusk, & J. Boyle, Automated Reasoning Introduction and Application, Prentice-Hall, Engelwood Cliffs, N.J., 1984

TUTORIAL SESSION

Expert Systems and Their Use in Nuclear Power Plants

ROBERT E. UHRIG
University of Tennessee
Knoxville, Tennessee 37996-2300, USA

and

Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6009, USA

The report is being published by Plenum Publishing Company
in
Advanced in Nuclear Science and Technology
(In Press) 1989

The review of Expert Systems that resulted in this paper was supported, in part, by the U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, under Department of Energy Interagency Agreement 1886-8085-2B. This support is gratefully acknowledge.

The views and opinions expressed are those of the author and do not necessarily reflect the criteria requirements and guidelines of the Nuclear Regulatory Commission or the views of the author's employers.

Knowledge Acquisition and Representation Tutorial

ELIZABETH H. GROUNDWATER
Technical Director, Intellient Technology Division
Science Applications International Corporation
1710 Goodridge Drive
McLean, Virginia 22101, USA

ABSTRACT

This tutorial describes methods for extracting knowledge from human sources and methods for representing that knowledge in Knowledge-Based Expert Systems. These tasks are performed by a Knowledge Engineer. Methods for extracting knowledge from human experts include interviewing (both structured and unstructured), task analysis or simulation (using familiar, constrained, or tough cases), and more formal, psychologically-based techniques. Knowledge from these methods can be collected in various ways, using videotapes, audiotapes, written notes, or formal data collection sheets.

Knowledge representations include forward or backward-chained if-then rules (the most common representation), object-oriented representations such as frames, relational representations such as semantic networks and hierarchies, and typicality representations such as scripts and stereotypes. Examples of each representation and some discussion of the types of problems to which each is suitable are included. The goal of the tutorial is to provide an overall understanding of the available methods and techniques and terminology associated with them, so readers can make informed decisions in selecting an approach to their own expert system development projects.

1.0 KNOWLEDGE ACQUISITION METHODS

Knowledge acquisition is considered to be a difficult assignment in AI and is also one of the critical parts of expert system development. The first step in development is the acquisition and characterization of the knowledge and skills of an expert. The identification and encoding of this knowledge is one of the most complex and arduous tasks encountered. For this reason, considerable effort must be invested in acquiring skill and

experience in the art of knowledge acquisition. It should be given great emphasis in the development of expert systems.

Knowledge acquisition is widely regarded as a bottleneck in the system development process. Therefore, it is essential that methodologies and techniques used to extract the knowledge needed are selected with care and wisdom. Without an efficient, effective, and practical approach, the common bottleneck problem would greatly hinder system development and could result in unbearable delays.

There are several important and proven methods for acquiring knowledge. These methods are listed in Table 1.1. In this section we will describe how these methods can be used in acquiring knowledge for an expert system.

1.1 Interviewing

There are two types of interviews that can be conducted with an expert, the unstructured interview which is the most prevalent, and the structured interview, which is the most useful. In an unstructured interview, the knowledge engineer asks more-or-less spontaneous questions of the expert while the expert is performing (or talking about) a familiar task. The development of most expert systems starts with unstructured interviews of the expert. They are particularly useful in gathering the initial data and serving as an "ice breaker" to establish the relationship between knowledge engineer and domain expert.

A major problem encountered with some expert systems is that developers have taken it for granted that an unstructured interview is the only way to extract expert knowledge. This is understandable when the developers have not formulated (or are not familiar with) any structured methodologies or techniques for knowledge elicitation. While unstructured interviews serve as an important part of the knowledge elicitation process, in most cases they should not be the only method used. A drawback of the unstructured interview is that it requires skill and discipline. It is difficult to keep the conversation from wandering and becoming unproductive, or to keep track of issues and what still needs to be covered. However, prepared lists, outlines or conversation plans can be used as references without formalizing

Table 1.1

KNOWLEDGE ACQUISITION METHODS

Method Category	Description
• Unstructured Interview	The expert is informally queried for knowledge of facts and procedures.
• Structured Interview	The expert is queried on specified facts and procedures. Answers may be format constrained.
• Method of Familiar Tasks	Observation of the tasks that the expert usually performs
• Method of Tough Cases	Observation of tasks that are difficult and challenging for the expert
• Limited-Information Tasks	A familiar task is performed, but the expert is not given certain information that is typically available.
• Constrained Processing Tasks	A familiar task is performed, but the expert must do so under time or other constraints.
• Goal Decomposition	The expert must break up the overall goal into sub-goals, and then decompose sub-goals.
• Rule Induction	The expert system is "trained" with a series of problems and answers and induces rules from the examples.

an unstructured interview and can provide helpful reminders to the interviewer. Note taking can also help to improve the discourse.

The structured interview is an interview that is guided by a structured format prepared beforehand by the knowledge engineer. It is typically performed after the first pass knowledge base has been created and addresses questions that arose from the first pass knowledge base. An important feature of this knowledge elicitation method is that it causes the experts to systematically go back over the knowledge they have previously given. This review has a number of predictable effects on the knowledge base including: 1) the addition or deletion of entries; 2) the qualification of entries; 3) the reorganization of the hierarchical or categorical structure of the knowledge base; or 4) the addition or deletion of categories. These various effects will generate a second pass knowledge base.

A potential problem with interviewing is that it only elicits knowledge of which the expert is consciously aware. Another drawback of interviewing is the potential for bias. If the knowledge engineer over "leads" the interviewing, then the elicited knowledge may be unduly constrained and/or distorted. This is particularly common when the knowledge engineer is above the novice level of the domain. In interviewing, the knowledge engineer constructs the questions, rather than having the expert generate all the information. Therefore, a non-novice interviewer could easily direct the interviewing in a direction that he considers to be important rather than allowing the domain expert to define the importance of issues.

An important tool in conducting interviews is note-taking on the part of the knowledge engineer. The interviewer should always take substantial on-the-spot notes even if he is also taping the interviews on video or audiotapes. Note-taking helps the knowledge engineer maintain an alert, inquiring attitude and to generate helpful questions.

1.2 Performance Observation

Performance observation involves watching the expert work on real or simulated problems. These problems may be familiar tasks, tough cases, tasks where the information is purposely limited, or constrained tasks. The

knowledge engineer avoids interrupting the expert during the task and debriefs the expert at the end.

If the expert is willing, protocol analysis is often used in performance observation. It requires the expert to verbalize his or her thought processes either during or after performing the task, though during is more informative. Verbal protocols are then analyzed to reveal how the task was performed. The information is usually captured by recording the session on audiotape.

The goal of performance observation is to collect information on the decision-making process. Data collected should include: 1) goals and sub-goals, 2) problem-solving steps, 3) decision points, 4) input and supporting data, and 4) products created. During the debriefing, the decision points should be reviewed to discover alternate paths or branches that could have been pursued if the task data was altered slightly. Performance observation, if performed well with a cooperative verbal expert can be the most powerful method of knowledge acquisition.

Within the area of performance observation, the method of familiar tasks involves studying the expert while, within the area of performance observation, he or she is engaged in usual or typical kinds of tasks. Looking across a set of situations of the experts' specific tactics and procedures, it is possible to find commonalities in goals, information the expert prefers to have available, and actions that are taken in response to various situations. This establishes a "baseline" for understanding how problems are solved by the expert. Familiar tasks are usually examined early in the knowledge acquisition process.

The method of tough cases is used to refine aspects of the expert's reasoning during the later phases of knowledge elicitation. Subtle or refined aspects of the expert's reasoning are often manifested when an expert encounters a tough case, a case with unusual, unfamiliar, or challenging features. Since tough cases are rare, the expert is not likely to encounter one while the knowledge engineer is present. If he cannot set aside the case for the next knowledge elicitation session, he should attempt to record a verbal protocol on tape as he works through the case.

Limited-information tasks are similar to familiar tasks, but the amount or kind of information that is available to the expert is restricted. This forces the expert to rely heavily upon (and hence provide additional evidence about) his knowledge and reasoning skills. Generally speaking, experts do not like it when information that is available to them is limited in some manner. However, once they are adapted to this "game" situation, they can provide a wealth of information. The limited information task is very useful in revealing the experts' reasoning and inference strategies (as opposed to factual knowledge). The incompleteness of the information available stimulates the formulation of hypotheses (rather than final judgments), strategic thinking (What if...?), and the use of heuristics when performing a task.

Constrained-processing tasks are like limited-information tasks in that both involve tinkering with the conditions for performing familiar tasks. Constrained-processing involves a deliberate attempt to constrain or alter the reasoning strategies that the expert uses. There are many techniques used to accomplish this, some of which are domain-specific. Examples include: 1) limit the amount of time that the expert has to absorb information or make judgments; and 2) ask the expert one or more specific questions rather than require the full analysis that is conducted during a familiar task. Constraints and limited - information conditions can be combined, if needed.

1.3 Structuring Methods

Structuring methods are based more on psychological theory than the previously described methods. They tend to provide more specialized and less diverse types of information, thus are less often useful. They are used to define categories, relationships and other groupings of knowledge elements versus the elements themselves. Two example methods are goal decomposition and similarity judgments.

Goal decomposition consists of breaking a problem (or goal) into sub-goals. It is a traditional problem reduction approach, and is useful for enumerating goal states and describing general categories of goals. The primary purpose of this method is to determine the overall structure of the problem, when a backward-chaining (goal-driven) reasoning approach seems

appropriate. Goal decomposition can be accomplished by having knowledge engineers query the experts about how they try to determine the cause of some problem. The answers can then be regarded as sub-goals that are further reduced until they cannot be broken down further, or the process becomes inefficient. Efforts to acquire detailed rules in this way have not been successful because goals express purposes but not detailed descriptions or interpretations of situations.

Similarity judgments can be used to refine the subtle points of the knowledge base where the rationale may be obscure. The method basically consists of posing two or more scenarios, events, or items to the expert and having him determine how they are similar or different. The expert should then explain why the chosen groupings and distinguishing traits are important. The similarity judgments can then be subjected to a multidimensional scaling analysis, if required. The information collected is often used to establish frame or object types and to divide a rule base into rule sets.

1.4 Rule Induction

The principle of rule induction is that the expert supplies a set of example cases with which to "train" the system. Instead of eliciting rules directly, the system builder infers them from the expert's performance on example problems. Thus, the emphasis is on how they do things as opposed to what they do.

The process begins by having the expert provide the relevant factors and attributes influencing a set of decisions. The system algorithm then uses the training set to induce general principles and formulate the decision process. This makes it possible to predict decisions for cases not contained in the example set. The major advantage is that experts often find it easier to provide examples of decision solutions than to describe the decision making process itself. Disadvantages include 1) irrelevant or meaningless rules may be induced, 2) the rules tend to be unstructured and difficult to understand, and 3) the expert may not choose sufficient cases or cases of the right types to correctly train the system.

Example of Rule Induction:

```
From:     (P a), (P b), ...
Infer:    (forall (x) (P x))

From:     (if  (inst leaf-1 leaf)                 We have one leaf
               (color leaf-1 green))              that is green
          (if  (inst leaf-2 leaf)                 We have a second
               (color leaf-2 green))              leaf that is green

Infer:    (forall (x) (if (inst x leaf)           Infer that all leaves
                          (color x green)))       are green
```

1.5 Method Efficiency

In a practical sense, method efficiency is an important criterion for selecting knowledge acquisition method(s) because scarce resources such as the domain expert's time are wasted otherwise. To measure method efficiency, one divides the number of knowledge propositions (rules, facts, whatever) gathered during a session by the overall time it took to prepare, conduct, and analyze the session. A rule of thumb is that a good session results in about two new propositions per minute, while a poor one results in less than one per minute. Usually a mix of methods versus focusing on any particular one produces the best results.

2.0 KNOWLEDGE REPRESENTATION METHODS

The goal of the knowledge acquisition process is to construct a representation of the knowledge that can be implemented in an expert system. There are several knowledge representation methods available for the knowledge engineer. They include rules, semantic networks, isa and ispart hierarchies, frames, object-oriented programming, conceptual dependency, scripts, and stereotypes.

2.1 Rules

Rules provide a formal way of representing recommendations, directives, or strategies expressed as "if premise then conclusion" or "if condition then action". The premise or condition is called the antecedent of the rule and the conclusion or action is called the consequent. The antecedent may contain several clauses linked by "AND" or "OR". The consequent may consist

of clauses or verb phrases. The consequent clauses would be asserted as true when the condition of the rule is met (this is called firing). The verb phrases would specify actions to take when the rule fires.

Because rules provide a very explicit, modular representation for knowledge, they are easily modified. A rule-based system is an expert system whose knowledge base is comprised primarily of rules. An example of a rule is shown below.

IF GOLD FALLS BELOW $250 PER OUNCE,
AND STOCKS ARE SHOWING AN AVERAGE GAIN OF 5% PER YEAR,
THEN INVEST 40% OF THE PORTFOLIO IN GOLD.

The antecedent of a rule may contain free variables. A free variable is one which has not yet been assigned a value. This assignment is known as binding. These variables are bound (set to) the objects which make the antecedent true. The inference engine tries "bindings" until the antecedent is true or all the bindings have been tried. These bindings remain for the consequent, allowing a rule to "fire" for many different data elements in the problem domain.

A rule-based system consists of three parts - a set of rules which comprise the rule-base, one or more databases which contain known facts or assertions, and an inference engine which specifies a strategy to determine in what order the rules should be examined and possibly fired or executed.

The database contains the facts or assertions that pertain to the domain. It may also include objects that are symbols that represent real world constructs such as a chair, desk, room, etc. These facts, as in the previous example, are matched against the clauses in the antecedent of the rules in the rule-base. When all of the antecedent clauses are matched, facts are modified or added to the database or some other action is taken.

The inference engine draws inferences by matching conditions with facts or other rules, deciding which rule to select if more than one is found appropriate, and firing selected rules when their conditions are met and which may entail adding new elements to the assertion list. How does the inference engine decide which rule to fire if more than one is applicable? That depends on the inference engine. Most do not allow the knowledge

engineer explicit control but do allow some implicit control. Common practices are to select the first rule matched, the most specific rule matched, an arbitrary rule matched, or the newest rule on the assertion list matched.

Rules are often grouped together that have common knowledge or domain or that solve common problems. These groupings are called rule sets. Proper use of rule sets increases the modularity and understandability of the overall system. Rule sets can be used to provide a level of abstraction by having each high-level goal solved by a different rule set.

There are many advantages to rules as a representation strategy. Rules are modular; they behave much like independent pieces of knowledge. Rules provide an easy way to represent problem solving knowledge of what to do in a specific situation. Rule-based systems are similar to the way people talk about how they solve problems. Lastly, rules provide a uniform structure for the knowledge in a rule-base. There are also disadvantages to rules as a representation strategy. The two most important are that the flow of control may be difficult to follow and it is difficult to represent complex objects using rules.

2.2 Semantic Networks

Semantic networks are concepts connected by relationships. They may be diagrammed as nodes connected by arcs. The arcs are labelled with the relationship between nodes. The nodes may represent any object or concept in the problem domain. This method of knowledge representation is attractive to many in the artificial intelligence field because they hypothesize that knowledge in the brain may be organized in this manner. Anatomical connections, which also resemble this network, bear no as-yet-known relationship to the organization of knowledge, however.

The advantages of semantic nets are that important associations are explicit, there are direct links between related pieces of information, and the method is very general. One disadvantage is that the meaning of the knowledge depends on the program manipulating the network. Other disadvantages are that inference may not be valid, and it is difficult to represent the passage of time.

2.3 Hierarchies

Hierarchies are a specialization of semantic networks. There are two types of hierarchies that are especially important. They are ISA and ISPART hierarchies. In an ISA hierarchy the items are classes and the relationship represented is the subclass relation. The exception is that the lowest items in the hierarchy may be members of the subclass, in which case they are related by the member, or instance, relationship. Subclasses inherit the properties of their superclass, although usually a means to override this feature is provided. ISA hierarchies are attractive because they are intuitively familiar; humans tend to classify. An example of an ISA hierarchy is shown below.

Class Properties

Animals Live
Birds Fly
Ostriches Can't Fly

ISA Hierarchy

Sam isa Sparrow
Fred isa Ostrich
Sparrow isa bird
Ostrich isa bird
Bird isa animal

Conclusions

Sam Flies
Sam Lives
Fred Can't Fly
Fred Lives

In an ISPART hierarchy, systems are broken down into their components. Those components may be subsystems which can in turn be broken down further. The relationship modelled in an ISPART hierarchy is "is a part of". An ISPART hierarchy example is shown below. Next to each item, in brackets, is the input and output of the item. Underneath each item is a list of the items that comprise it.

Nuclear Power Plant [input: U 235 output: electricity]
{Primary system, Secondary System, Power Generating system, Control System}

Primary system [input: U 235   output: heat to steam generator]
{Nuclear reactor vessel, primary coolant pump, steam generator, pipes, valves, fluid(water)}

Secondary system [input: water, heat from Primary   output:steam]
{steam generator, secondary feedwater pump, pipes, valves, fluid (water and steam)}

Power Generating system [input: steam   output: electricity, water]
{turbine, electric generator, condenser, pipes,valves}

2.4 Frames

Frames are object-oriented data structures. They can be considered a specialization of ISA hierarchies because inheritance is the most heavily utilized relationship between the frames. Frames are used to represent classes of objects. The subclass and superclass relations are supported between frames. Individual objects are instances of a class represented by a frame. A frame is the collection of attributes for the object type.

Slots are used to represent the attributes of a frame. The slot will contain the value of an attribute at a particular time. Slots may be attached to procedures, so that when the value of the slot is desired, a particular procedure is called to calculate it. Slots may also be set externally, (e.g., by a rule). The value of a slot might be an object. In this case the slot name is the relationship to the object and frames and slots together make up a semantic net. An example of some frames and their slots is shown below.

```
Animal Frame
     Metabolic Rate:
     Life Span:
     Subframes:Fish,Mammal

Mammal Frame
     Super Frame:Animal
     Mate:
     Children:call findchildren
     Subframes:Dog, Cat

Dog Frame
     Superframe:Mammal
     Favorite Dogfood:
```

If Charlie is an instance of the frame, Dog, then its particular frame might follow the next example. Notice that Charlie has inherited all of the slots of all the subclasses to which he belongs.

```
Charlie
    Metabolic Rate: 100 calories/hour
    Life Span: 10 yrs
    Mate: Sheila
    Children:{Bob, Spot}
    Favorite Dogfood: Alpo
```

In developing a frame structure, the knowledge engineer has considerable flexibility. One of the dimensions of that flexibility is whether to have a deep or shallow frame hierarchy. A deep frame structure has many ancestors between the instances and the topmost frame, while a shallow frame structure has few ancestors. The choice that the knowledge engineer has to make is whether an attribute should be a slot or whether objects with different values of the attribute should be in separate subclasses. This can be summed up by "one man's slot is another man's class."

How does a knowledge engineer decide between making an attribute a slot or using it for different subclasses? He should consider any attribute of the object which may cause objects to be in different classes. If varying this attribute will cause the system to treat the object differently, then the attribute should be used to form separate classes. Otherwise, the attribute should be made a slot. For example, an expert system that determines depreciation won't treat different makes or types of automobiles differently, but one which schedules vehicles for tasks might treat trucks very differently from cars.

There are many advantages to using frames. Complex objects, especially those that contain many attributes, are easy to represent. Inheritance is easily represented. Frames are very flexible and therefore good when the calculation of attribute values is complex. A disadvantage of frames is that their meaning or function is often not as clear as a simple production rule. As such, it is relatively easy to use frames incorrectly.

2.5 Object-Oriented Programming

In object-oriented programming the focus (as the name implies) is on objects. In this programming paradigm, many independent objects are created, with each object having its own (possibly unique) means of communicating and interacting with other objects in the system. Each object behaves with other objects in the system, storing and manipulating data in its own private section of memory, and determines its own behavior in response to what other objects in the system are doing.

This data or object-centered approach to programming is implemented by asking objects to perform operations on themselves instead of passing data to procedures. An operation on an object is instigated by a message passed to that object, requesting it to perform the operation on itself. Operations with the same conceptual meanings will have the same name (or identifier) but will be implemented by the objects in different ways.

An example of the object-oriented approach may be derived from a nuclear power application. In this example, two objects are 1) NUCLEAR REACTOR, and 2) PRIMARY COOLANT LOOP. The operation to be performed is SLOW DOWN (x%). The objects are requested to perform (via a message) the SLOW DOWN (x%) operation. This operation has the same conceptual meaning for both objects, but will be implemented differently by each object. The NUCLEAR REACTOR will insert control rods at the appropriate distance, whereas the PRIMARY COOLANT LOOP will decrease its water flow by some x%.

There are four distinguishing principles of object-oriented programming, all of which must be satisfied by the programming language to be truly object oriented. These principles are 1) information hiding, 2) data-abstraction, 3) dynamic binding, and 4) inheritance.

The information hiding principle dictates that the state of a software module is contained in private variables that are usable only from within the scope of the module. External modules can access a given module only through an explicitly designed module interface. Information hiding is important for ensuring reliability and modifiability of software systems by reducing interdependencies between software components.

Data abstraction can be considered a method of using information hiding. In data abstraction, high-level operations are employed which hide (in a manner similar to information hiding) the details of both the operations and the underlying data structure on which the operation is performed. A common example of this is the use of the addition symbol (+) to add either integers or real numbers. Dynamic binding requires that the actual procedure associated with the operator is not determined until its use at runtime. This principle is necessary to fully implement the data abstraction principle.

Inheritance enables programmers to create classes and hierarchies of objects. Properties of a class (e.g., maximum weight) are passed (i.e., inherited) to subclasses, which may in turn be passed to further subclasses, etc. In a similar manner, properties of members of a class may be inherited to describe specific instances of a class. Inheritance allows a programmer to create new classes of objects by specifying only the differences between a new class and an existing class. As new classes do not have to be specified from scratch, a large amount of code can be reused.

Messages are the means by which objects communicate. A sample message is the SLOW DOWN (x%) request above. It is the responsibility of the receiving object to determine what action to take in response to a message and what reply to provide for the sender of the message. The action which an object takes in response to a message is called a method. A method manipulates the information stored in the private memory of the object and may send messages to other objects in order to accomplish its task.

As noted above the classes of objects in the object-oriented environment may be thought of as templates for the creation of a specific set of objects by the programmer. The act of creating an object using an existing object as a template is called instantiation, and the object that is created is considered to be an instance of the template from which it is derived.

The object-oriented and frame-based approaches are very similar. A frame-based approach uses many of the elements of an object-oriented approach. In a frame-based system, frames may correspond to either classes or instances and methods are implemented by procedures or rules. Both

systems have class hierarchies (both with varying levels of sophistication) and both have slots for describing object attributes. The principal difference between the object-oriented and frame-based approach is that in a true object-oriented system the only way to get information about an object or to modify the information an object possess is through a message. This is in contrast to most frame-based expert systems where slot values may be modified directly. The frame-based approach thus lacks many of the advantages arising from the information hiding and data abstraction principles of the object-oriented approach.

Advantages of object-oriented programming include 1) a reduction in program size through the use of hierarchies resulting in lessened development and maintenance costs, 2) high modularity and reusability of code by using large libraries of pre-existing class and object definitions, 3) the clarity of expression of object-oriented code, and 4) the gain in software reliability gained by the clarity of code and reduced program size. Disadvantages of the object-oriented approach are 1) increased execution time due to overhead caused by the dynamic binding and messaging, 2) the additional time necessary for coping with an unfamiliar programming technique, and 3) the use of a pre-existing class library requires a significant learning period to use that library effectively.

2.6 Conceptual Dependency

Conceptual dependency diagrams are used for natural language understanding. In conceptual dependency, a representation of a sentence is built from conceptual primitives. Usually there are 4 types of primitives roughly corresponding to objects (nouns), actions (verbs), object modifiers (adjectives), and action modifiers (adverbs). An example of an action primitive is ATRANS, the transfer of an abstract relationship. For example the verb, sell, signifies the transfer of the abstract relationship, ownership. The primitives are chosen to be independent of any language so that the conceptual dependency representation of a sentence is independent of the original language. An advantage of this representation is that inferences can be drawn which are, themselves, not dependent on the language of the statements from which they are derived.

2.7 Scripts

Scripts are a description of stereotypical sequences of events. They are activated when their preconditions are satisfied. They are especially useful when there are a limited number of scenarios to consider. It is possible to derive a lot of high-level information from a few facts when using a script. An example is shown below. Notice that from two simple facts we are able to derive many conclusions, two of which are shown.

o Preconditions:	x walks into a restaurant, r
o Script:	y is Host(ess) of r y seats x y hands x r's menu x gives order x receives order x eats order
o Facts:	Sam is Host of The Ritz Fred walks into The Ritz
o Conclusions:	Fred sees The Ritz's menu After script, Fred has eaten

2.8 Stereotypes

Stereotypes are clusters of characteristics often found together in an object. They are often used in reference to people. An example of when stereotypes might be useful is in implementing an intelligent machine interface. If the system can conclude that user is a new user (from command patterns or erroneous commands and keystrokes) then it can conclude from the stereotype that he will need more prompting and explanation.

REFERENCES

1. Hayes-Roth, Frederick; Waterman, Donald A.; and Lenat, Douglas B. Building Expert Systems, 1983, Addison-Wesley Publishing Co., Inc., Reading, Massachussetts.

2. Hoffman, Robert R. "The Problem of Extracting the Knowledge of Experts", AI Magazine, Volume 8, Number 2, Summer, 1987, pp. 53-68.

3. The EPRI Knowledge Acquisition Workshop Handbook, EPRI NP-6240, February 1989, Electric Power Research Institute, Palo Alto, California.

4. Waterman, Donald A. A Guide to Expert Systems, 1986, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts.

5. Wolfgram, Deborah D.; Dear, Teresa J.; and Galbraith, Craig S. Expert Systems for the Technical Professional, 1987, John Wiley & Sons, Inc., New York, New York.

A Tutorial on Real-Time Expert Systems

HOWARD ROSENOF, ROBERT MOORE, GREGORY STANLEY, and RICHARD SMITH
Gensym Corporation
125 Cambridge Park Drive
Cambridge, Massachusetts 02140, USA

ABSTRACT

The practical application of expert systems to dynamic domains requires a new approach toward knowledge representation and application. In particular there is a need to represent dynamic qualitative knowledge, dynamic analytic knowledge and the structure of the object interactions. The application of inference in real-time requires paradigms which go beyond pattern matching, which for example use metaknowledge to focus the inferencing resources of the expert system and which may be readily integrated with other systems. Finally the application of truth maintenance requires a temporal model of the time dependence of the truth of data and of inferred results.

The real-time expert system allows the representation of deep knowledge, both analytic and heuristic. Graphic and structured natural language interfaces allow the user to construct knowledge bases for dynamic applications, to test expert system behavior under dynamic conditions and to validate knowledge bases under various dynamic scenarios. The interactive development interface allows short development-and-test cycles. Built-in bidirectional data interface facilities allow the engineer to implement an application interactive with sources of data and actuators of commands.

KNOWLEDGE REPRESENTATION

Expert systems represent the knowledge of the human expert more explicitly than do conventional computer programs. In an expert system the human expert's knowledge is represented separately from

the representation of the solution, whereas in conventional software the two are closely intermingled. The human expert's knowledge may then be applied by the expert system to a variety of problems, without the extensive reworking that would be required by conventional code. The expert system is also generally easier to maintain, since changes to the problem-solving strategy and to the description of a problem are largely independent. Knowledge representation refers to the representation of the human expert's problem-solving strategy as well as knowledge of the problem.

REAL-TIME CONSIDERATIONS

For an expert system intended to operate in real-time, several characteristics of dynamic domains impose requirements on the knowledge representation:

1. A task scheduler is required, so that tasks such as rule antecedent evaluation, display updates and data acquisition can all be prioritized and executed without waiting for the completion of other tasks.

2. The concurrent use of analytic and heuristic models. Conventional simulation methods allow analytic models (e.g. differential equations). Conventional expert systems allow heuristics (rules) and leave the analytic part for the user to program. The combination of directly-represented analytic and heuristic knowledge, in an object oriented framework, allows the applications to be addressed in a unified way.

3. Interaction between objects. The structure of an application is frequently important in predicting behavior, performing diagnosis and in scenario simulation. Structure is generally expressed within the class hierarchy, and as as connectedness of objects, or proximity of objects. Structure may also be expressed in an object's attributes, especially where relations may vary in time. An object-oriented framework which has the built-in capability to reason in terms of object connectedness or proximity, and which can integrate analytic as well as heuristic knowledge in these terms, allows efficient representation of the structural knowledge about an application. In industry, connectedness is particularly useful in simulations and in real-time process analysis, since in both, the

behavior of some item of equipment depends on the states of the items connected to it by pipes, wires, etc. Connectedness is useful as well in other network-oriented problem representations, such as occur frequently in scheduling applications. Here the connection can represent a sequence relationship.

4. Dynamic behavior and live data. Many problems have a real-time aspect, including dynamic knowledge in differential equation form, such as equations of motion. Live data may be needed for the expert system's eventual deployment, and timely data access and real-time processing may be important. Conclusions may be based not only on the instantaneous value of some variable but on its past behavior as well. A framework which includes these real-time considerations in the expert system design is required. This framework allows simulation to provide time-varying values for prototyping and development, to be supplanted by sensor-based data at deployment. Data servers provide interfaces to other applications with a minimum of user work, so the prototype can become the actual application.

5. In addition to applications which often require a unified framework, the general desirability of rapid implementation calls for the use of high level interfaces to the human expert. These may include graphical construction of the application domain, and structured natural language for expression of knowledge, models, and other information. Modern parsing techniques allow the user to express the knowledge in a reasonably natural form.

APPLICATIONS

Diagnostics:

Early applications of real-time expert systems have followed the pattern established by the earlier static systems: The first well-known expert system (MYCIN, 1972) was built to automate the analysis of medical data and propose a diagnosis and a corresponding course of treatment. Early applications of expert systems to the process industries attempted similarly to analyze process data and direct the attention of plant personnel to the most likely cause of some problem. It is possible to build such an application as a static expert system, conducting a dialog with plant personnel in much the same way that MYCIN conducts a dialog with a physician. However,

these applications are today being supplanted by real-time systems which can automate the acquisition of information by conducting a "dialog" with control equipment, SCADA (Supervisory Control and Data Acquisition) equipment, and so forth. Also, real-time expert systems can base their conclusions and recommendations on history; calculating and using time rates-of-change, for example.

Diagnostic applications are of great interest to expert system builders because they are so difficult to construct by conventional means. There is little serious work being done in computer-based diagnostics, that does not involve the techniques of artificial intelligence. This is also the case, albeit to a lesser extent, for the other major application area of static expert systems, planning, scheduling and optimization.

Planning, Scheduling and Optimization

Mathematical optimization techniques (linear programming, nonlinear programming, etc.) are common, but expert system techniques have gained a foothold, particularly in cases where the mathematical techniques are too expensive. For example, one approach to optimization is to use conventional techniques to automatically generate all possible solutions to a problem, and then use expert-system techniques to reject most of them. The evaluation of candidate solutions, which is computationally expensive, is thereby limited to those which have a reasonably high likelihood of being selected as optimum.

Some scheduling problems are not readily amenable to analytic techniques at all. For these, expert systems may provide the only practical way to automate the preparation of schedules. In other cases where mathematical techniques are usable but expensive, it may be more economical to use an expert system frequently, providing schedules that are not genuinely optimal but are good, over mathematical techniques which are restricted to only occasional use. Finally there is the possibility of using the expert system to schedule the optimization, so that the latter is run only when the operational improvements it will probably lead to, can justify the cost of the optimization.

Simulation techniques are often used to evaluate the effects of proposed policies. Here the simulation may be built within the expert system, or the expert system used as the end-user interface to a simulation implemented externally.

Real-time expert systems have an advantage over static systems (and analytic techniques as well) when schedules and policies need to be changed because of rapidly-changing process conditions. One example is in the navigation of autonomous robots, where resources (the robots) must be deployed in accordance with frequently-changing criteria. Another example is optimizing around unscheduled outages of equipment. A third is in contention logic, where in real-time, a scarce resource may be allocated among contending users, in accordance with realistic optimization criteria rather than time-of-request or arbitrary order.

As a practical example of expert-system optimization, one company is building a heuristic tuner to optimize a particular process by adjusting the control over individual loops, in a coordinated fashion.

Control

Real-time expert systems add a new control language to ladder logic, block diagrams and the other established varieties; and in fact may be readily extended to include most of these as well (as long as analytical representations are allowed to co-exist with heuristic representations). Actuator control largely continues to be exercised by conventional techniques, but expert systems are finding increasing application, especially in supervisory control. (Optimization may be considered as one form of supervisory control). This is true in a variety of settings, but particularly where autonomous operation is needed and conventional control methods cannot readily cope with the variety of conditions to which the controller must respond. In one example, expert system techniques provide the sole control over a modular water-desalinization system, designed to operate for long periods of time in areas isolated from normal maintenance support.

General Applications

Starting in a domain in which there are few if any practical alternatives, real-time expert system technology has moved progressively into domains which are served by other software technologies. The unique advantages of this technology are applicable to uses far beyond the "traditional" domains of artificial intelligence. Today the real-time expert system should be evaluated as as a possible development and delivery environment for virtually any application requiring automated responses to changing data. Even this criterion, however, does not represent the practical limits to the application of this technology: In part because of their advanced end-user interface capabilities real-time expert systems are used as intelligent front-ends to applications such as simulators (and are also used as simulators themselves). In one application, a real-time expert system was selected to provide high-level interfacing between dissimilar applications, almost solely because of its excellent communication capabilities.

KNOWLEDGE BASE MANAGEMENT

Design techniques for development of conventional software emphasize the organization of code to facilitate development and maintenance. Knowledge-based systems also benefit from proper organization. In the latter case an organization based on hierarchial windows, or "workspaces" may replace one based on calls to modules of code. Workspaces may be associated with objects or object definitions, and these may in turn have subworkspaces which themselves have objects, rules, etc., without any fixed limit on the number of levels in the hierarchy. The knowledge base may to an extent me managed by workspaces; for example, by activating all rules on a particular workspace.

Immediate availability of changes to the application (no compile step) and knowledge-base access tools that can display user-specified items while the application is running, are useful development tools.

REASONING ABOUT KNOWLEDGE

Two apparently conflicting requirements dominate the inference paradigm considerations in the real-time domains. One is the need for truth maintenance. With values of thousands of variables subject to frequent change, the validities of conclusions at all levels of

inference are in question. The other requirement is for real-time performance, which implies response fast enough to advise the human operator in time for the operator to take some action, and/or directly exercise some level of control over the process.

First attempts at using expert systems for real-time applications involved taking a "snap-shot" of data and using a static expert system paradigm to perform inference. This process is then repeated after inferences are completed. Conventional pattern-matching paradigms which examine all possible conclusions for the current data values are, however, too slow for many applications of interest. The static expert system approaches led to slow performance on even small prototypes of a few hundred rules and a few hundred data. In addition to the overhead of drawing many conclusions, a communications bottleneck can arise. Out of thousands of variables that might be available, only a few may be neded at any one time. This has been widely recognized.

Code improvements and computer improvements can help. However a fundamentally different inference approach has been found generally appropriate for real-time applications. This emulates the approach that a human expert uses in a real-time situation: Maintain a peripheral awareness across the domain, watch for performance exceptions, and then focus on areas of interest. The inference engine in the real-time expert system continually scans knowledge which the expert has specified for peripheral awareness. If a safety-threatening condition occurs in a reactor, for example, the inference engine uses metaknowledge to determine which additional knowledge to invoke, thus focusing on the area of interest. The inference engine may apply knowledge appropriate to any number of concurrent problems, with timeliness of results maintained subject to limitations of the hardware.

One benefit of the metaknowledge approach is that very large knowledge bases can be run in real time. Since many types of problems and behaviors are represented in the knowledge base, it can get quite large, with thousands of rules. However the inference engine should not consume computer time looking for patterns in all of this knowledge all the time. Rather it focuses attention on the knowledge needed. The concept is like the human thought process, in that a human does not use knowledge of swimming or driving when walking in

the park. The human mind focuses, using the knowledge relevant to the task.

A characteristic of many real-time tasks in generating stations and other large plants, as well as in geographically dispersed networks such as for electric power transmission, is distribution of knowledge. Specialized human operators may exert control over parts of some domain, cooperating with others on issues of mutual concern. It is appropriate for the expert systems that assist them to similarly maintain local autonomy while cooperating as needed. This practice can yield expert systems with limited scopes of knowledge each, making them more maintainable, and can help system developers manage the interactions between the constituents of the domain.

In static expert systems, truth maintenance involves changing inferences when data changes. In real-time problems there is an additional requirement to change inferences even if no new data is available, since time is a factor in validity or certainty of inference. One way to express this temporal validity information is to attach an expiration time to each value maintained by the inference engine, and propagate this when inference is carried forward. Generally, when a conclusion is based on several time sensitive variables, the earliest of their respective expiration times will be carried forward. Expiration times can be propagated forward through multiple levels of inference, but there are also ways to limit this propagation when appropriate.

TASK SCHEDULING

Real-time operating systems assign resources to tasks in accordance with the tasks' priorities and outside influences. A real-time expert system must exist as a set of tasks available for scheduling by the real-time operating system, or may exist as one task ("process" in terminology common to operating systems) if it has its own scheduling functions. In the latter case, it is possible for the expert system to deliver usable real-time performance while running under the control of a conventional, non-real-time operating system.

In many applications the real-time expert system must be able to accept unsolicited messages and reschedule its task processing to

accommodate them. This is particularly true of those systems performing process monitoring and fault diagnosis functions.

SUMMARY

The real-time expert system technology described in this paper represents a departure from static expert system design, as the issues of time relationships and dynamic behavior have been addressed. The resulting expert system is capable of applying thousands of rule-frames of knowledge, and of performance in real time for reasonably complex operations. Installations of the technology include chemical process plants, manufacturing and robotics.

REFERENCES

1. J. S. BYRD, J. J. FISHER, and W. R. MALLET, "Expert Robots for Process Environments," Robotics and Autonomous Systems, Elsevier Science Publishers, New York, N.Y. (to be published 1989).

2. R.L MOORE., L.B. HAWKINSON, C.G. KNICKERBOCKER and L.M. CHURCHMAN, "A Real-Time Expert System for Process Control," Proc. First Conference on Artificial Intelligence Applications (IEEE), Denver, CO, 529, (1984).

3. R. L. MOORE and M.A. KRAMER, "Expert Systems in On-Line Process Control," Proc. Third International Conference on Chemical Process Control, Asilomar, CA, (1986).

4. B. G. BUCHANAN and EDWARD H. SHORTLIFFE, "Rule-Based Expert Systems", ISBN 0-201-10172-6, Addison-Wesley Publishing Company, 1985.

Tutorial on Validation and Verification of Knowledge-based Systems

LANCE A. MILLER
AI Technologies
Defense Systems Group
Science Applications International Corporation
McLean, Virginia 22102, USA

ABSTRACT:

The Validation and Verification (V&V) of Knowledge-Based Systems (KBS) becomes of increasing concern as these systems are fielded and embedded in the everyday operations of Electric Power and other industries -- particularly so when life-critical and environment-critical aspects are involved.

We provide in this tutorial a V&V perspective on the nature of KBS components, an appropriate life-cycle, and applicable testing approaches. We consider why KBS V&V may be both easier and harder than traditional means, and we conclude with a series of practical V&V guidelines.

1. OVERVIEW

The problem of assuring correctness of knowledge-based systems (KBS) and expert systems (terms used interchangeably here) is a good deal more critical for the utility industry than most other areas in that the ultimate consequences of failures of this AI technology, when accompanied by other technology breakdowns, can be nothing less than disastrous for, particularly, our nuclear plants -- and much of the AI activity is in these facilities.

Consider just two examples of KBS, REALM (the Reactor Emergency Action Level Monitor; 41) and the EOP (Emergency Operating Procedures) Tracking System (31). Both systems have modes which operate in real-time for emergency situations: REALM provides an emergency director with an analysis of input data and classification of the emergency conditions and designation of the appropriate Emergency Action levels; the EOP Tracking System similarly analyzes nuclear plant conditions and identifies the appropriate emergency procedures to support the operators' decision-making processes. Both systems have been painstakingly designed, implemented, and tested, but still we have to be concerned, as for any piece of software, about some lurking as-yet-undetected problem in the code which under just those rare right conditions will produce critically deceptive errors.

The Electric Power Research Institute has long been active in promoting research into KBS for utility applications, and they very early recognized the need for addressing the V&V issues and supported seminal work in this area (35, 36). This tutorial builds upon this and other previous work and focuses on what V&V is all about, how it ties in with KBS development life-cycles, what the various elements are to be tested, why KBS V&V may be both harder and easier than for traditional systems, and what kinds of testing

techniques can be applied. We conclude, finally, with a collection of recommendations for practically implementing V&V.

2. V&V DEFINITION AND LIFE-CYCLES FOR KBS

2.1 Definitions.

The terms Validation and Verification suffer confusion among themselves and with related terms like "testing" and "evaluation." Perhaps the clearest definitions are given in IEEE Std. 729-1983 (18):

Verification The process of determining whether or not the products of a given phase of software development meet all the requirements established during the previous phase.

Validation The process of evaluating software at the end of the development process to ensure compliance with software requirements.

Many Department of Defense (DoD) projects involve the life-cycle of DOD-STD-2167 consisting of, usually, the following 5 stages: Requirements Definition, Design, Development, System and Acceptance Test, and Operational Support (and General Support). For this life-cycle, Verification would occur 4 times.

Validation, strictly speaking, occurs only once -- at the end of the last stage of the life-cycle -- at which time the final system is compared to the original requirements. The objective of Validation is not to insure the accuracy of each development stage but rather to insure that, however developed, the final product is and does exactly as it was intended in the beginning. Validation can be simulated in earlier stages to the extent that one can devise tests which address the question "If a real system were developed and delivered according to the specifications developed so far, would this system satisfy the customer's initial requirements?".

Given these explications Boehm's (5) succinct summary makes very good sense:

Verification "Am I building the product right?"

Validation "Am I building the right product?"

2.2 Related Standards and Concepts.

The Federal Information Processing Standard (FIPS) Publication 101 (14) provides a number of guidelines for V&V of standard procedural software, but many of the precepts apply equally as well to KBS. NUREGs 0696 and 0737 (29, 30) provide clear requirements for V&V within the nuclear utility industry for emergency response facilities and capabilities.

Concerning individual life-cycle stages, ANSI/IEEE Standard 830-1984 (1) contains an excellent characterization of the problems and needs of good requirements specifications. The comparable publication for system design is ANSI/IEEE Standard 1016-1987 (4). Standard 1008-1987 (2) provides guidance

for software test plans and procedures, while 1012-1986 (3) provides a comprehensive coverage of developing an overall V&V plan.

A related term to V&V is that of **Certification**. This usually denotes an activity following Validation where the system's robustness is assessed under real (or simulated) field conditions (11). Other related terms are those of **testing, evaluation, quality assurance,** etc.; all these techniques are concerned with assessing software quality, functionality, and performance, and they all can be used for the narrower focus of V&V (6).

2.3 KBS Life Cycle.

The main problem with applying the above standards and guidelines to KBS is that they have a much different life-cycle than that for which the standards were designed. Some workers have adopted Boehm's (5) spiral model of 4 successive activities (requirements, design, implementation, and operation/test) continuously being repeated and refined (37). Others (10, 8) have proposed a 4-stage model of Problem Definition, Initial Prototype Stage, Expanded Prototype Stage, and Delivery/Maintenance Stage.

Consistent with both of these life-cycles is one which the author and others at SAIC have employed in building expert systems. Shown in Figure 1, it is an elaboration on a model constructed earlier for EPRI which reviewed and developed KBS V&V methodology for Nuclear Power Plant Applications (28, 35, 36).

The Figure 4 life-cycle has 3 main stages: Development, Integration, and Maintenance. There are 2 primary aspects of the initial Development Step: development of the Initial Prototype, and development of the Revised Prototype. The initial prototype provides a partial but working model to provide a demo of some of the more important requirements, particularly the operational concept, that can then be evaluated with the customer and potential users. The key result is that requirements can be much better understood and revised accordingly, while elements of the initial prototype may or may not be retained. The Revised Prototype consists of an iterative series of prototype developments until an evaluation indicates that it is adequate.

Across the two development prototypes are 4 phases of V&V and five associated documents. Although these phases are called **V&V**, the primary activity is that of Verification; but Validation could occur through simulation, as discussed previously. The initial V&V phase, called V&V Phase 0, is to gather and record preliminary goals and requirements of the system during the early discussions with the customer and the initial exploration of requirements. The document recording these findings is the Preliminary Expert Systems Requirements Specification (P-ESRS), and it is the beginning reference document for V&V.

V&V Phase 1 occurs at the first step of the Revised Prototype, when the requirements are revised, based on the evaluation of the Initial Prototype. Customers quite commonly initiate development of KBS or Expert Systems with one set of requirements in mind and then change these radically as they work with the initial prototype. Their revised requirements are detailed at this step and documented in the Expert System Requirements Specification. The Phase 1 V&V process is to compare the initial set of requirements, as

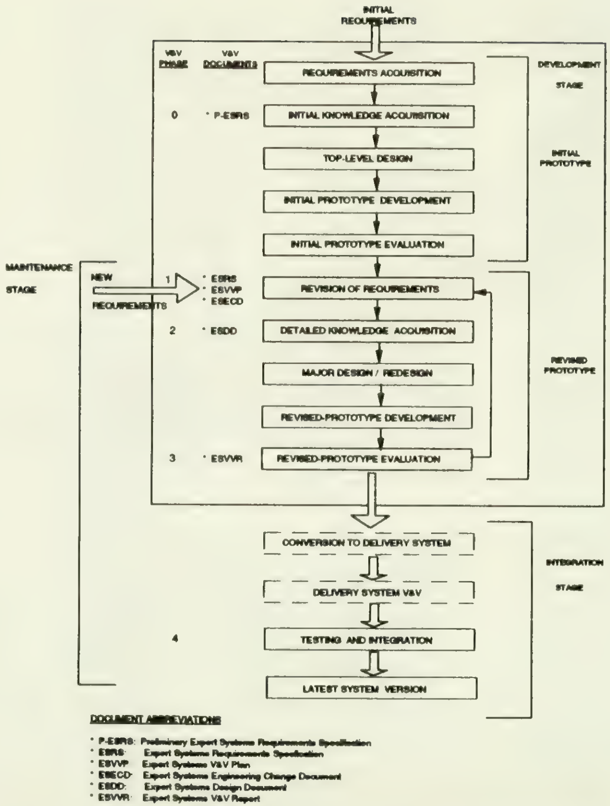


Figure 1: Life-cycle For The Development Of Knowledge-based Expert Systems (Courtesy Of Science Applications International Corporation)

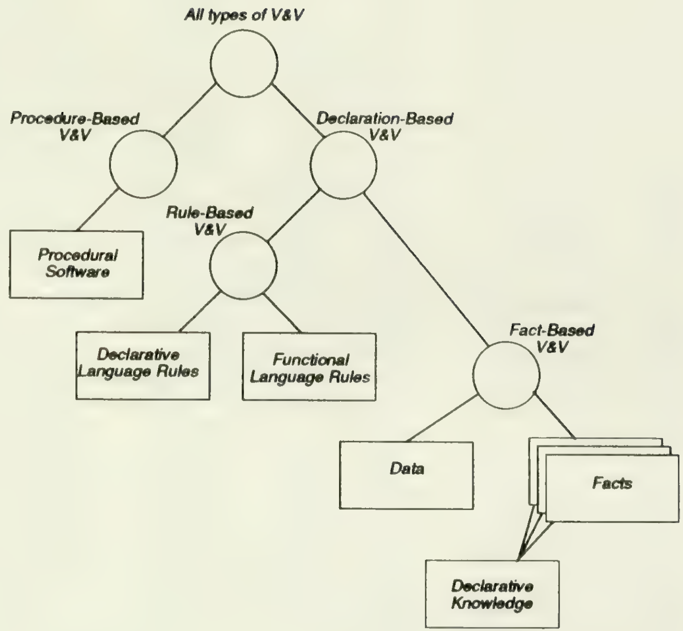


Figure 2: The various types of V&V (circles) and their associated objects of investigation (boxes).

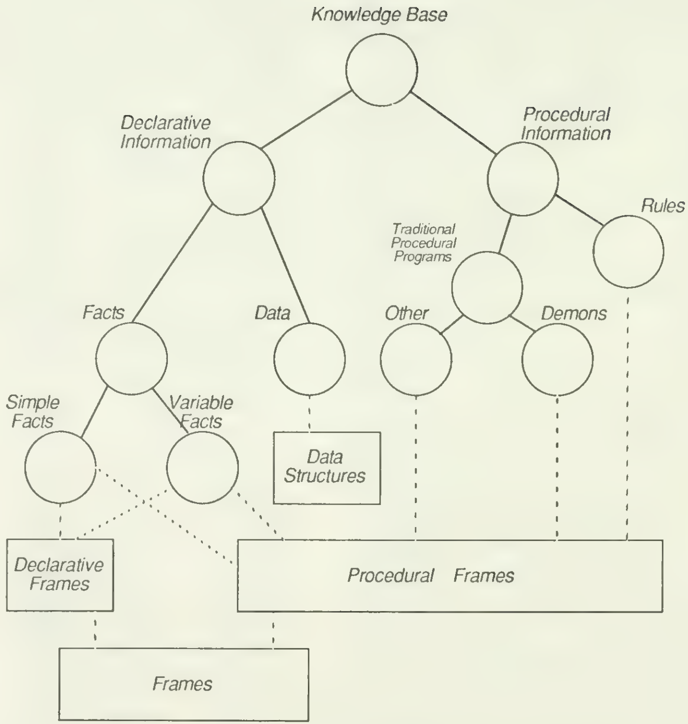


Figure 3: Various types of information in a Knowledge Base (circles) and the information structures which contain them (boxes)

	SIMPLE	COMPLEX
Interactivtiy with other systems	Stand alone consulting expert system	embedded real time interrupt driven
Source of Knowledge	Codified (written)	elicited multi-source (experts!)
Search Space	simple/small (large but easily factorable)	Very Very large Unfactorable Complex dimensionality
Type of Inferencing	Simple backward or forward chaining no uncertainty	truth-maintenance constraint-based reasoning fuzzy reasoning or reasoning under uncertainty

Figure 4: Four factors of complexity in KBS and Expert Systems

expressed in the P-ESRS with these new requirements as documented in the ESRS; this activity will assure that all of the preliminary concerns initially expressed by the customer and potential users are either accurately represented in the new set of requirements or else withdrawn. The ESRS is modified as a result of this Phase 1 V&V process.

An important component of the ESRS is the specific measurements and acceptable performance values to be applied to each requirement -- e.g., rather than writing a response-time requirement as "The system shall respond in a timely manner", the range of acceptable seconds of delay for various kinds of input tasks is to be specified. This information is a primary input for the second major document which is best produced at this point -- the Expert System V&V Plan (ESVVP). This plan will document the procedures by which the Revised Prototype is ultimately to be tested against the specifications in the ESRS.

V&V Phase 2 occurs at the second step of the Revised Prototype, when the major design has been accomplished and documented in the Expert System Design Document (ESDD). The design provides a clear exposition and justification of the architecture and the detailed plan needed to implement the ESRS requirements. The V&V process is to compare the design against the requirements (the ESDD with the ESRS) to assure that it adequately addresses all of the requirements.

V&V Phase 3 occurs at the last, evaluation, step of the Revised Prototype development. The prototype is first tested against the design document (ESDD) for Verification, and then the prototype is Validated against the original requirements as contained in the ESRS.

Following Development is an Integration phase for converting (if need be) the Revised Prototype into a delivery-system language or format, and then integrating that system into its embedding system(s) to provide the latest collective version of the overall system. V&V Phase 4 then occurs at the Testing and Integration step of this process by repeating the Validation tests for the now-integrated system against the original requirements.

The last major step, Maintenance, provides for changes, fixes, or introduction of new requirements into the start of the Revised Prototype development. A final sixth document is used to record the details and circumstances of each of these modifications -- the Expert Systems Engineering Change Document (ESECD). Then the ESRS is updated to reflect these changes.

This V&V life-cycle process thus specifies the steps in development at which V&V should occur, along with the associated documents to be produced and used. What it doesn't provide is a clear statement of <u>what</u> should be tested and <u>how</u>. These are covered in the next 2 sections.

3. THE GENERAL V&V CHARACTERISTICS OF EXPERT SYSTEMS

3.1 <u>Definition of KBSs</u>.

KBS are sometimes defined in terms of their components:

> The major components of an expert system are knowledge base, inference engine, user interface mechanism (including explanation facility), and data ... (17)

or, more commonly, in terms of their particular functionality:

> Expert systems are computer systems, comprising both hardware and software that mimic an expert's thought processes to solve complex problems in a given field (domain). (42)

While we might agree with the latter emphasis on intelligent function, from the V&V point of view it is only the first of these definitions that is really pertinent. The V&V of expert systems is different from that for traditional programming systems <u>because the nature of the components is different</u>; an inference engine plus a knowledge base makes for a greatly different testing problem than for the normal procedural code of Ada, FORTRAN, or COBOL (see section 3.3).

A corollary is that -- **for V&V purposes only** -- a KBS originally developed in an AI language but subsequently rewritten in a procedural language is then no longer an KBS type of V&V problem; it is a traditional V&V situation.

3.2 <u>Two Types of V&V</u>.

In accordance with the above definition of expert systems, we necessarily must distinguish two types of V&V. The first we call <u>Procedure-Based V&V</u>, and it involves the evaluation of, first of all, **software** which, second, is **procedural** in nature. It is Procedure-Based V&V which has received the most attention, particularly by DoD, and about which there has been the most development of standards, tools, and methods.

The second type of V&V is concerned with non-procedural KBS elements; we call this second type <u>Declaration-Based V&V</u>. This type is concerned with the knowledge in the KBS (see below), either the simple declarative-statement type knowledge of **data, facts, and declarative frames** -- for <u>Fact-Based V&V</u>-- or else with the **IF-THEN rules** of the knowledge base -- for <u>Rule-Based V&V</u>. Figure 2 shows all of these types of V&V. The reason for distinguishing these various types is that various testing techniques are better for one type than another (see section 4).

3.3 <u>Components of KBS</u>.

Table 1 shows the major components of expert systems. The key V&V concern, given the previous section, is whether the component involves traditional procedural code or something else.

3.3.1 <u>Knowledge Base</u>. The elements of the Knowledge Base require some explanation. The **Rule Base** consists of all those items which are logically equivalent to **IF-THEN** statements of the form **IF x THEN y** (possibly with an additional **ELSE** z clause) in which **x** is one or more Boolean-connected assertions which can each be evaluated to either **True** or **False**, and in which **y** (and z) are either assertions of fact or else the equivalent of calls to procedures which cause some action to be taken.

Simple Facts are assertions that the value of some specified attribute has a particular specific value; two examples are

time(Clock1, 1324) (Clock1 shows 1:24 pm o'clock) and

weight(Henry, 174) (Henry weighs 174 pounds)

Variable Facts are Simple Facts in which at least one of the Fact constants is replaced with a variable, indicating that its value is unknown (or at least as yet unrestricted), such as:

time(x2, 0923) Time on clock **x2** is 9:23 a.m.

weight(x3, x4) A man **x3** weighs **x4** pounds.

Data refers to values of attributes which occur in some well-defined data structure or database. Whereas Facts always specify a particular attribute (or relation), and constants or variables are supplied as arguments to this attribute-expression, Data attributes are not immediately associated with the actual values; they are derivable from knowledge of the semantics of the Data Structure -- such as the fact that, for a **Table** of data, all of the values in a particular column have the same attribute, namely that specified at the head of the column.

From a V&V point of view, Facts, being more explicit, are going to be easier to V&V than Data. Fact reasoning involves explicit statements of facts and rules which utilize these facts to make some deduction or assertion, as in:

Facts: age(Bob,21), person(Bob)

Rule: adult(x1) <-- person(x1) and age(x1,x2)

and greater(x2,20)

--> Because Bob is 21 years old, Bob is an adult.

Data reasoning always involves the possibility that, since the Data are pure numbers or character strings, unlike attribute-value Facts, there is greater opportunity for using semantically incorrect Data in the reasoning.

A final type of non-procedural information is that of **Declarative Knowledge**. As used here Declarative Knowledge refers to an organized collection of Facts (simple and complex) such that the collection constitutes a <u>definition</u> of some entity, attribute (or relation), or action concept. The familiar notion of a "frame" is based on Declarative Knowledge. When only Declarative Knowledge is included in a Frame, it is called a Declarative Frame. Frames can also involve Rules (rare) as well as some special small programs called "Demons" (common). **Demons** are procedures which are to be executed under certain value conditions -- when the value is unknown, when its value is suspect, etc. Finally, Frames could also contain calls to special assembly-language (or other) programs, or to procedural functions. Frames containing Demons, Rules, or any procedural software are called Procedural Frames, and V&V of this type of frame will require both Procedure-Based and Declaration-Based V&V.

Table 1: Components of KBS and Expert Systems

Major Component	Subcomponents [1]
Knowledge Base	Rule Base Simple Facts Associated DataBases Frames Demons Procedures, Functions
Shell [2]	Utilities User interface Knowledge Representation Facilities
Inference Engine	pattern matcher decision procedure(s) rule-ordering procedures
External Interfaces	Operating System DB Management System I/O Devices (User Interface) Programming Language Environments Other Applications Utilities, functions, procedure calls

[1] List is not necessarily exhaustive
[2] A Shell is not a mandatory component

Table 2: Criteria to be tested or evaluated for three major classes of requirements

Class	Criteria	Class	Criteria
Functionality	consistency completeness/coverage correctness number of solutions error-handling/recovery explanation help/tutoring Meta-Knowledge	General Atrributes	portability maintainability reliability availability usability security safety fault tolerance flexibility/extendability understandability cost
Performance	Execution Speed Memory Requirements I/O Handling Power (re human-time to solution, quality, success, experience level) Efficiency Database access time time-to-solution		

Table 3: The applicability of various testing procedures to components of Expert Systems[1]

APPLICABLE TESTING METHODOLOGY [2]	Primary V&V Type [3]	COMPONENTS OF KBS AND EXPERT SYSTEMS						
		KNOWLEDGE BASE				Inference Engine	External Interfaces (incl shell)	Whole System
		Rules	Frames	Simple Facts	Data Bases			
DYNAMIC TESTING								
Random Test Selection (19)	P, D	3	5	5	1	2	1	1
Functional Testing (16)	P, D	3	5	5	2	1	1	1
Structural Testing (33)	P	5	5	5	2	1	1	1
Regression Testing (12)	P, D	4	4	5	5	3	5	5
Explanation (7)	D	1	1	1	4	5	5	2
Simulation (32)	P, D	1	5	5	5	3	1	1
COMPETENCY TESTING								
"Gold Standard" (34)	P	5	5	5	5	4	4	1
Effectiveness Procedures (24)	P	5	5	5	5	4	4	1
Workplace Averages (34)	P	5	5	5	5	4	4	1
STATIC TESTING								
Anomaly Detection[4] (38)	R	1	2	1	3	5	5	5
Structured Walk Throughs (13)	A	1	1	2	5	1	1	2
Informed Panel Inspection (10)	A	1	2	2	5	3	3	2
Clean Room Inspection (27)	P	2	3	5	5	1	1	2
Mathematical Verification (20)	P	3	5	5	5	3	3	3
Symbolic Execution (21)	P	4	4	5	5	3	5	5
Well-formedness (26)	F	2	1	1	3	5	5	5
Fault-Tree Analysis (22)	P	2	5	5	5	2	2	1
Automated Data-Flow Analysis (11)								

[1] The lower the numerical value of the entry, the more the judged appropriateness (scale from 1 to 5)
[2] A reference to a desription of some methodologies is given in parentheses
[3] P = Procedure-Based V&V, D = Declaration-Based V&V, F = Fact-Based V&V, R = Rule-Based V&V, A = All types of V&V
[4] Includes all problems identified in Table 4

Table 4: Classes of Knowledge-base rule anomalies. (adapted from Stachowitz and Coombs, 1987)

TYPE OF CHECK	PROBLEM	RULE EXAMPLE/EXPLANATION
Structure	Dead-end nodes	For rule A(x) and B(X), there is no fact, goal, or schema which matches A(X)
	Unreachable nodes	For fact Father(x), there is no rule which has a left-hand side (LHS) which references such a fact.
	Redundancy: Duplication	One rule is redundant: A(X) and B(X) ->C(X), B(X) and A(X)->C(X)
	Redundancy: Subsumption	hot(X) and tired(X) ->unhappy(X), hot(X) -> unhappy(X)
	Cycle: Direct	ancestor(a,b) and parent(b,c) -> ancestor (a,c)
	Cycle: Indirect	human(X) -> person(X), person(X) -> human(X)
	Inconsistency: Direct	tall(X) -> heavy(X), tall(X) -> ¬ heavy(X)
	Inconsistency: Indirect	tree(X) -> plant(X), elm(X) -> tree(X) elm(X) -> ¬ plant(X)
	Irrelevance	wet(X) and rich(X) ->uncomfortable(X), wet(X) and ¬ rich(X) -> uncomfortable(X)
Semantic	Incompleteness	car(X) and GT(Speed(X), 70) ->unsafe_speed(X) car(X) and LT (Speed(X), 30) ->safe_speed(X) ?? GE (Speed(X), 30) and LE (Speed(X), 70)
	Out-of-range	FACTS child(Adam), age(Adam, 20), Person(Adam) RULE person(X) and age(X, Y) and LE(Y, 12) -> child(X)
	Illegal value	Legal_Value (Project_Status, <on_time, behind ahead>) Project(X), Project_Status (X, late)
	Incompatability	setting up an association of the values of argument with the value of another (and then violating the association -- values_of_college_year (<freshman, sophmore, junior, senior) Values_of_freshman_courses (<101, 102, 103>) FACTS Freshman(X), course (x, 201)
	Incorrect subrelation	Specify that "murder" is done to a person whereas "kill" is done to any living thing -- Kill (X, bush) is OK but murder (X, bush) is not.

The relationships among all the types of information in a Knowledge Base, and the associated structures, are illustrated in Figure 2.

3.3.2. Shell. While a Shell is found in many expert system development products, it is not a necessary component; PROLOG, for example doesn't have a shell. Shells are frequently written in assembler or other procedural language and therefore must be examined by Procedure-Based V&V tools.

3.3.3. Inference Engine. Procedural code, e.g. fast-executing assembly-language software, is also typically involved with the Inference Engine whose 3 primary subcomponents are listed in Table 1. The **pattern matcher** is responsible for matching up facts with rule elements, as well as matching the conclusion of rules (the so-called "right-hand sides" of IF-THEN expressions) with other rule elements. In the Fact Reasoning example of the previous section, the pattern matcher would be responsible for matching **age(Bob,21)** with **age(x1,x2)** and for asserting a temporary "binding" between the names **Bob** and **x1** as well as between **21** and **x2**.

The **decision procedure** refers to aspects of searching a goal space. The primary types of search are bottom-up vs top-down and depth-first vs. breadth-first. Bottom-up search begins with facts, proceeds to associate facts with the left-hand sides (LHSs) of IF-THEN rules, assert their consequences (right-hand sides, RHSs) as elements of LHSs of new rules, etc. Synonymous with bottom-up are the phrases "data-driven" and "forward-chaining." Top-down procedures begin with the statement of some goal (a RHS of some rule), attempt to instantiate or bind values to the LHS of the rules found, and so work backwards to the Fact elements; a synonym is "backward-chaining." In terms of rules, breadth-first refers to making a substitution or binding to matched patterns in all rules first; depth-first means taking the first rule in which there was a match and trying to work from that particular rule either forwards or backwards. The native decision-procedure in PROLOG, for example, is almost always top-down depth-first.

Combinations and variations of these kinds of search processes underlie more complicated types of inferences such as **constraint-based** or **truth-maintenance** (pruning search spaces based upon specific variable-binding) or **opportunistic reasoning** (pursing alternately forward- and backward-chaining to speed up lining Facts to ultimate goals; used in means-ends search). The decision procedure is thus an extremely complicated and critical component of the Inference Engine.

The **rule-ordering procedure** is a method for deciding which rule is to be examined next when the pattern matcher and decision procedure jointly identify two or more possibilities.

3.3.4 External Interfaces. The External Interfaces usually connect procedural code to the expert system usually via internal procedural code. These concerns collectively are therefore usually those of Procedure-Based V&V.

3.4 KBS Complexity Factors.

There are four main aspects of complexity of expert systems, as shown in Figure 4. Any one instance of complexity is sufficient to make the task of V&V difficult; occurrence of 2 or more should alert the tester to maximum

diligence in executing his V&V plan and urging the developers to follow every possible good development practice.

Interactivity is complex when the expert system is embedded with other programs such that the KBS exchanges information with them. The most complex kind of interactivity is for real-time interrupt-driven systems. These can be exceedingly complex for a wide variety of reasons (11), but expert systems bring some additional problems: non-deterministic backtracking in logic-based programs interferes with scheduling of event processing, and the declarative nature of rule-based programs may not directly associate with the real-time events/interrupts the appropriate response (which therefore has to be deduced).

Concerning the knowledge-source, when knowledge is codified, formulaic, and written down, the development of rules in a KBS is usually a straightforward routine task. When it has to be elicited from people, it's a much more difficult task. If the knowledge needed is that of an expert, or-- worse -- distributed among several experts -- then one must count on an extended and tortuous knowledge elicitation and verification stage.

The third complexity factor concerns the search space. Relatively small spaces can be exhaustively searched, if needed, by brute-force methods. Large spaces for which efficient factoring or elimination procedure exist also pose little problem. But very very large spaces whose dimensionality is not known or well-understood and for which there are no known search algorithms can cause great difficulties developing adequate KBS.

Finally, concerning inferencing, most existing KBS involve either data-driven problems like diagnosis, which involve forward-chaining, or are planning systems which begin with a clear goal and use backwards-chaining to achieve the solution; these are relatively well-understood and pose little difficulty. But either a complicated form of reasoning -- such as truth-maintenance or constraint-based -- or a form of reasoning under uncertainty or with fuzzy values can cause great difficulty in achieving a finally stable and successful system.

As one reviews the opportunities for KBS in, particularly, the electric power utility industry, they invariably contain a large number of problems which would be called complex according to the above criteria. Effective **and trusted** fielding of these systems will rely greatly on thorough V&V of them.

4. KBS TESTING TECHNIQUES

In testing a KBS one has to be concerned with three questions: (1) **what component** of the KBS is to be tested; (2) for **what criteria** is to be tested; and (3) what **testing procedure** should be used. The answer(s) to the last question depend critically, of course, on the response to the first two questions; there is no technique or procedure which is useful across all components and criteria.

4.1 Testing Criteria.

A major guide to the criteria for which one should test KBS (and any other system) is given by ANSI/IEEE standard STD 830-1984 (1). These have

been added to and adapted to form the three classes of criteria shown in Table 2. In terms of the components of KBS identified previously (see Table 1), the following rough generalizations can be made: **Functionality Criteria** apply primarily to the Knowledge Base; **Performance** and **General Attribute** criteria apply to the other components.

The first three entries for Functionality criteria are taken here to comprise a subclass for which the testing technique is termed **Anomaly Detection**, discussed in detail in section 4.3.

Explanation refers to the capability of the system to provide rationale for the functional processing it performed -- usually in terms of a trace of the IF-THEN rule reasoning it followed. In life-critical systems, weapon-systems, intelligence-processing applications, and any installation in which the consequences of incorrect action are severe, an adequate Explanation facility will typically be a major requirement.

The last functionality criterion, **Meta-Knowledge**, refers to whether or not the system has explicit and accessible knowledge concerning what it "knows", what it can do and can't, what its goals are. In an intelligent data-base application for example, Meta-Knowledge would provide for a useful answer to a user's query "What things do you know about?"; it would also permit it to give an instructional response to the query "How many female telephones are installed at this installation?" by informing the user that **gender** was not an attribute of telephones.

4.2 Overview of Testing Methodologies.

Table 3 shows the applicability of 3 classes of testing procedures to the KBS components. **Dynamic Testing** primarily involves operating the system and either (1) treating the system as a "black box", evaluating the output for selected input combinations, relative to the requirements, or (2) opening up the system and tracing or otherwise following internal execution of processing elements (sometimes called "white-box" testing).

Competency Testing is a special case of the above techniques in which the performance of the system is compared to external capability of other types of systems -- usually involving human operators. Finally, **Static Testing** involves a non-operating examination of the system's code or knowledge.

The techniques covered by each of these classes are briefly discussed in the following sections (access to the technical literature on each is provided by the references shown in the table).

4.2.1. Dynamic Testing Methods. With the exception of Explanation, these methods were all developed for traditional procedural software. However, most can be usefully applied to KBS as well, at least at the system/interface level.

Random test generation has been shown to be an extremely effective means of discovering system errors in traditional systems. **Functional** Testing involves developing tests for each separate function or functional requirement of the system and systematically providing complete coverage of these. **Structural Testing** is concerned with generating tests to exercise the

structures within the software, almost always from the perspective of <u>control</u> structures and paths (data flows are certainly equally as important, but effective data flow analysis programs for test construction are not yet widely available).

Regression Testing involves repeated use of previous test-case suites periodically as changes are made to the system. This is a particularly effective procedure when an incremental development process is employed -- as is often the case with KBS development.

Explanation, for V&V purposes, is a special type of tracing application in which the KBS is instrumented to provide rule-trace explanations throughout execution.

Finally, **Simulation**, when applied to the exercising of developed KBS, involves the generation of realistic combinations of data input patterns for the KBS and is particularly useful for embedded real-time KBS involving substantial data input.

4.2.2. <u>Competency Testing</u>. These procedures compare the black-box performance of the KBS to external standards.

The **Gold Standard** refers to a performance standard that has been widely agreed upon by interested parties as anything from minimum acceptable levels to optimum. **Effectiveness Procedures** involve no pre-established standard but rather various methodologies (e.g., double-blinds) by which knowledgeable persons assess the quality of system performance. **Workplace Average** procedures derive standards of average performance by identifying the performance levels of human operators performing under conditions comparable to the KBS.

Competence tests, as applied to KBS, are useful really only for whole system evaluation, not for its components.

4.2.3 <u>Static Testing</u>. With these techniques the system is opened up for anatomical and inferred performance inspections; they are primarily useful for the Rules and Frames components.

Anomaly Detection is the most developed and important of these techniques and is discussed separately in the next section. **Structured Walk Throughs** involve a group inspection of code or knowledge, usually led by the developer, with the group representing all of the interested roles (maintainer, user, manager, installer, etc.). The idea of the **Informed Panel Inspection** is similar -- to assemble KBS experts familiar with the intricacies of, particularly, the reasoning processes embodied in the system (as well as the complex environment).

Clean Room Inspection is on the informal end of formal mathematical inspection, designed for procedural code, while **Mathematical Verification** is at the formal end. Symbolic Execution, also a mathematical approach, involves trying to solve the data assignments in the code as equations, comparing these to known data constraints.

Well-formedness refers to an approach by the author for determining the validity of Frame knowledge by assessing whether Frames are "well-formed", according to a set of criteria; well-formedness in turn is associated with

increasing probability of validity and broader knowledge coverage.

Fault-Tree Analysis was designed especially for life- or system-critical testing, emphasizing detection and avoidance of failure modes somewhat over true functionality. **Automated Data-Flow Analysis**, best represented by the problematic L/PSA tool, focuses on data relationships and has special value in its emphasis on modeling -- and tracing -- requirements.

4.3 Anomaly Detection.

This approach, best represented by Lockheed's EVA system (37, 38, 39, 40) focuses primarily on the Rule Base but can be extended to other Knowledge components. EVA is, without question, the most important existing KBS V&V tool both for its conceptual contribution of functional anomaly classifications as well as its practical performance as an automated tool. Although it is not yet widely available, its processing can be hand-simulated, and the essential value to others is in its identification of the numerous problem classes.

The two problem-classes EVA identifies are shown in Table 4, and an example or explanation is provided for each.

Most of the Structure anomalies are intended to be self-explanatory, and only two are discussed here. In **indirect inconsistency** a contradiction is established by 3 or more rules in, potentially, non-obvious means. The example has an explicit rule that if X is an Elm it is not a plant; but, by the provision of the inferencing being closed under implication, X is shown first to be a tree if it is an Elm, and then -- by implication --it is a plant, a contradiction. **Incompleteness checking** will discover when a range of a variable (or value subset) is not covered by the rules; the example shows the system requesting how to classify car speeds of 30 up to 70.

The **Semantic** anomalies are concerned with discrepancies between facts and the restrictions on fact-values as given by rules. Although this testing approach is rated only a 3 for databases in Table 3, it is clear that this Semantic checking capability, if extended to databases, could be a very powerful tool for assuring their accuracy.

5. WHY V&V WITH KBS IS BOTH HARDER AND EASIER

Considering the practical aspects of implementing KBS V&V, the case can be made that V&V for KBS is both harder and easier than for traditional systems!

5.1 KBS V&V is Harder.

Some have been concerned about the typical KBS not involving traceable, testable requirements, as well as the fact that there is often not a single correct "right" answer to the problems addressed (9). Others have focused on the style of KBS programmers, seeing the documentation of requirements as being too constraining given the normal fast and dynamic pace of KBS prototype development (15). Others have worried about the unavailability of support tools for the KBS process, the fact that the logic and reasoning involved is often much more complex than in traditional programs, and

whether an initial requirements specification from the customer is realistically possible (35, 36).

We add our concerns about Knowledge being represented shallowly or insufficiently, lack of modularity in rule bases, lack of an established life-cycle culture, the difficulty of establishing a priori test plans and criteria values, and backtracking indeterminacy causing difficulties in performance assessments. We are also concerned that it's often very difficult to achieve well-bounded domains of the problem, and we worry about the problem of **elicited knowledge**, particularly from experts -- how are these sources themselves to be validated?

5.2 KBS V&V is Easier.

On the other hand, we have a much better chance of being successful in applying formal error-detection tests to the key code of most KBS, the rule bases, than ever of doing so for procedural code because the knowledge is kept separate from the control aspects of the code in KBS (39). Declarative knowledge is far easier to check even when accommodating uncertainty factors, and there are very promising opportunities for automatic program generation as well as for application of improved software engineering techniques to the KBS development process (23).

We would add that the usual incremental development process and the very tight feedback loop between prototype development and prototype assessment provides a much better means for achieving ultimate Validation than traditional development -- even when the latter involves prototyping, because of the difficulty of generating (and testing) drivers for prototype code. We also see strong forces towards representing knowledge more as frames than as rules (e.g., the DoD CALS initiative -- Computer-aided Acquisition and Logistics Support -- for electronic representation of process/product information); we believe frames are even more amenable to successful error-checking than are rules (26). Finally, we believe that, just as higher-level languages have made traditional programming so much more productive and of high quality, so will declarative knowledge or logic programming improve the art even more, by mapping so much more directly onto our cognitive processes.

6. PRACTICAL V&V RECOMMENDATIONS

Both of the above views on the difficulty of KBS V&V are, of course, correct. The need is to develop and use both management and testing techniques to guard against the negative aspects and capitalize on the positive. The recommendations to accomplish this are given below, organized in terms of the development process of Figure 1, preceded by some general points. By following these guidelines KBS can be fielded with high confidence in their reliability, much greater programmer productivity over all, higher system quality, and with much better maintainability expectations.

6.1 General Recommendations.

The major decision that has to be made is to truly commit to a serious V&V methodology -- at the outset of the project. Because Validation cannot be accomplished in any formal sense without a requirements specification,

this document -- something corresponding to the Expert System requirement Specification (ESRS) of the Figure 1 process -- must at least be obtained, even if nothing else is. Conceivably, for very simple systems involving no critical functions -- systems which are the least complex in terms of the 4 factors in Figure 4 -- this might just be sufficient, given that the design and testing are highly constrained and "obvious." For all other systems, however it is strongly recommended that documents corresponding to those 5 V&V development documents be developed and used as described in section 2.

Of course, following a V&V plan means following a specific development plan, so a serious commitment to V&V is also one for following a specific development life-cycle plan.

Given such commitments, probably the most generally effective thing one can do is to decide to follow the principles of good Software Engineering (11, 34). At the general level, all of these apply equally as well to KBS as to traditional software development, and there is absolutely no reason for not applying them seriously. Probably the single most important principle is that of **modularity** -- insuring that all design/implementation aspects of all the KBS component partitioning into subcomponents are as internally cohesive and homogeneous, and loosely inter-coupled, as possible.

Finally, one should anticipate and resist the seductive ease with which one typically can start-up and modify KBS prototypes, without thought of an overall plan or the ultimate resources needed to follow through. One should rather plan to use the same serious management scheduling and resource planning techniques as for any traditional project, and one should select the most powerful computer tool environment available to support all phases of development.

6.2 Requirements Specification.

Developing an agreed-upon written specification of system requirements is the single most important V&V task. The requirements should be **specific** and **testable** -- they are definite about what has to be accomplished for what system components, and explicit about the methods and criteria for testing, including minimum competence levels; tradeoff requirements (e.g., speed vs. accuracy) should also be prioritized.

A very important aspect to be emphasized about requirements is their relation to data -- what is used, referenced, accessed, modified and transmitted.

Finally, the requirements need to be thoroughly reviewed separately by the development team, and then with the customer and intended users, for **adequacy**, **accuracy**, and **attainability**.

6.3 Knowledge Acquisition and Representation.

Whatever techniques are used for acquisition, the most common development problem that impacts V&V is that of insufficiency of knowledge. One effective technique involves first organizing the knowledge required to meet the requirements into hierarchical modules, acquiring the knowledge believed necessary for each module, and then systematically extending the knowledge a bit beyond by asking of the sources for each item "What other cases-values-rules-conditions-components-processes could be involved for this

situation?". Also often overlooked are the means by which states of the world are assessed, particularly by experts. Thus, an expert's principle of "stop the presses if the paper is too thin" should be accompanied by the explicit method by which 'thinness' is measured, as well as what 'too thin' means.

Representation techniques which control the style of rule and fact/frame writing, follow good modularity standards, and minimize use of procedural code (like demons) will all positively impact the chances of successful V&V.

6.4 Design.

The most powerful (and CASE-supported) design technique, for both the declarative as well as the procedural aspects of KBS, is that of hierarchical data-flow specification (6, 11, 34). This approach focuses on identifying data stores and data transfers and is very effective for translating requirements into design aspects. However, the major utility for KBS is that it provides a coherent architecture for guiding the development of the specific rules (as well as frames) needed to implement the data-processes.

Key principles to be observed during design are those which anticipate and allow for activities later in the life-cycle -- i.e., designing for maintainability and modifiability. Additional important principles are designing for reusability of code and knowledge elements, for robustness in the face of noisy or unanticipated input, and providing special capability for error-avoidance.

6.5 Prototype Development.

The best two things one can do to insure good subsequent V&V during implementation are to provide a powerful development environment and to institute good configuration management and level control.

6.6 Testing.

This is naturally the main activity of V&V. Design Verification, if based on data-flow diagrams vs. requirements, is a paper vs. paper comparison involving any of a variety of Static techniques. Prototype testing will quite properly involve a number of the techniques shown in Table 3. **Dynamic Testing** should always be applied at least to the whole system, as well as to all interface and procedural components. Of the varieties, **Random Testing** and **Functional Testing** are the most effective (Structural Testing is generally believed to be of less value, and, if used, should explore only the simple control paths). **Regression Testing** should always be performed for each prototype version to assure that prior capability is retained.

Static Testing is properly applied to the Knowledge Base, and the most powerful approach is that of **Anomaly Detection**, which should always be applied to all rules. Inspection are most appropriate for Facts and Frames. The relevance of **Competency Testing** will pretty much be determined by whether the requirements specify performance criteria of this type or not, or whether the quality to be attained was very difficult to quantify.

Despite the plethora of testing techniques, if good V&V practices have been followed during development, the actual testing -- given Table 3-- could rather be a routine procedure, almost anticlimactic.

BIBLIOGRAPHY

(1) ANSI/IEEE Standard 830-1984, "IEEE Guide to Software Requirements Specification," IEEE Standard Boards, New York, NY, 1984.

(2) ANSI/IEEE Standard 1008-1987, "IEEE Standard for Software Unit Testing," IEEE Standards Board, New York, NY, 1986.

(3) ANSI/IEEE Standard 1012-1986, "Standard for Software Verification and Validation Plans," IEEE Standards Board, New York, NY, 1986.

(4) ANSI/IEEE Standard 1016-1987, "Recommended Practice for Software Design Description," IEEE Standards Board, New York, NY, 1987.

(5) Boehm, B.W., "A Spiral Model of Software Development and Enhancement," IEEE Computer, Vol 21, No. 5, May 1988, 61-72.

(6) Bruce, P. and S. Pederson, The Software Development Project: Planning and Management. New York: John Wiley and Sons, 1982.

(7) Buchanan, B.G. and E.H. Shortliffe, Rule-Based Expert Systems, The Myan Experiments of the Stanford Heuristic Programming Project, Reading, MA: Addison-Wesley, 1984.

(8) Citrenbaum, R.L. and J.R. Geissman, "A practical cost-conscious Expert System Development Methodology," Proceedings of the AI-1986 Artificial Intelligence and Advanced Computer Technology Conference, Long Beach, CA, April 1986.

(9) Culbert, C., G. Riley, and R.T. Savely, "An Expert Systems development methodology which supports verification and validation," Unpublished paper submitted for presentation at the Fourth IEEE Conference on Artificial Intelligence Applications, September 1987.

(10) Culbert, C., G. Riley, and R.T. Savely, "Approaches to the Verification of Rule-Based Expert Systems," Proceedings of First Annual Workshop on Space Operations Automation and Robotics Conference (SOAR '87), Houston, Texas, 1987.

(11) Deutsch, M. Software Verificaiton and Validation: Realistic Project Approaches. Englewood Cliffs, NJ: Prentice-Hall Inc., 1982.

(12) Downs, T. "An approach to the modeling of software testing with some applications." IEEE Transactions on Software Engineering, SE-11(4), April 1985.

(13) Fagan, M.E., "Advances in software inspection," IEEE Transactions on Software Engineering, SE-12(7):744-751, July 1986.

(14) Federal Information Processing Standards (FIPS) Publication 101, "Guideline for Lifecycle Validation, Verification, and Testing of Computer Software," 1983.

(15) Green, C.J.R., "On the Use of Requirements in Development of Knowledge-Based Systems." In Validation and Testing Knowledge-Based System Workshop,

20 August 1988. Given at the 1988 Conference of American Association for the Advancement of Artificial Intelligence, St. Paul, MN.

(16) Howden, W.E., "Functional program testing," IEEE Transactions on Software Engineering, SE-6(2):162-169, March 1980.

(17) Hu, S.D., Expert Systems for Software Engineers and Managers. New York: Chapman and Hall, 1987.

(18) IEEE Standard Glossary for Software Engineering Terminology, IEEE Std. 729-1983, Los Alamitos, CA, 1983.

(19) Ince, D.C., "The automatic generation of test data," Computer Journal, 30(1):63-69, February 1987.

(20) Jones, T.C., Programming Productivity. New York: McGraw Hill, 1986.

(21) King, J.C., "Symbolic execution and program testing," Communications of the ACM, 19(7):385-394, July 1976.

(22) Leveson, N.G., and P.R. Harvey, "Analyzing software safety," IEEE Transactions on Software Engineering, SE-9(5):569-579, September 1983.

(23) Linden, T.A., "Alternative Approaches to V&V for AI Systems." Proceedings from the Validation and Testing Knowledge-Based Systems Workshop at the 1988 American Association for Artificial Intelligence Conference, St. Paul, MN, August 20 1988.

(24) Llinas, J., S. Rizzi, and M.McCown, "The test and evaluation process for Knowledge-Based Systems," SAIC final contract report, Contract #: F30602-85-G-0313 (Task 86-001-01), prepared for Rome Air Development Center, 1987.

(25) Miller, L.A., "Knowledge Acquisition and Formal Representation of Requirements, Specifications, and Application Knowledge." Presented at the Second Annual Knowledge-Based Software Assistant Conference, August 18-20 1987, Utica, NY.

(26) Miller, L.A., "New Challenges for the Validation and Verification of Knowledge-Based Systems." AIAA presentation at session on "Issues in Validation of Knowledge-Based Systems," Reno, NV, 10 January 1989.

(27) Mills, H.D., M. Dyer and R. Linger, "Cleanroom software engineering," IEEE Software, 4(5):19-25, September 1987.

(28) Naser, J.A., "Nuclear Power Plant Expert System Verification and Validation." In Proceedings from the Validation and Testing Knowledge-Based Systems Workshop at the 1988 American Association for Artificial Intelligence Conference, St. Paul, MN, August 20 1988.

(29) Nuclear Regulatory Commission (NRC) Report, "Functional Criteria for Emergency Response Facilities," NUREG-0696, February 1981.

(30) Nuclear Regulatory Commission (NRC) Report, "Requirements for Emergency Response Capability," NUREG-0737, Supplement 1, December 1982.

(31) Petrick, W. and K.B. Ng, "Emergency Operating Procedures Tracking

System," EPRI NP-5250M, June 1987.

(32) Pritsker, A.A.B., Introduction to Simulation and SLAMII. New York: John Wiley and Sons, 1986.

(33) Rapps, S. and E.J. Weyuker, "Selecting software test data using data flow information," IEEE Transactions on Software Engineering, SE-11(4):367-375, April 1985.

(34) Rushby, J., Quality Measures and Assurance for AI Software. NASA Contractor Report #4187, prepared for Langley Research Center under contract NAS1-17067, October 1988.

(35) Science Applications International Corporation (SAIC), "Approaches to the Verification of Expert Systems for Nuclear Power Plants." Electric Power Research Institute Report NP-5236 (J. Naser, EPRI Products Manager), July 1987, prepared by Science Applications International Corporation.

(36) Science Applications International Corporation (SAIC), "Verification and Validation of Expert Systems for Nuclear Power Plant Applications." EPRI Report NP-5978, Project 2582-6, Final Report (J. Naser, EPRI Project Manager), August 1988.

(37) Stachowitz, R.A., and J.B. Combs, "Validation of Expert Systems," Proceedings of the Twentieth Hawaii International Conference on System Sciences, Kona, Hawaii, January 1987.

(38) Stachowitz, R.A., C.L. Chang and J.B. Combs, "Performance Evaluation of Knowledge-Based Systems," (keynote address) Aerospace Applications of Artificial Intelligence Conference, Dayton, Ohio, October 1987.

(39) Stachowitz, R.A., C.L. Chang, T. Stock, and J.B. Combs, "Building Validation Tools for Knowledge-Based Systems," Proceedings of First Annual Workshop on Space Operations Automation and Robotics, Houston, Texas, August 1987.

(40) Stachowitz, R.A., J.B. Combs, and C.L. Chang, "Validation of Knowledge-Based Systems," Proceedings of AIAA/NASA/USAF Symposium on Automation, Robotics, and Advanced Computing, Arlington, VA, March 1987. Also as: Chapter 3 in Machine Intelligence for Aerospace Systems, (eds., H. Lum and E. Heer); and in, Progress Series in Astronautics and Aeronautics, Vol 101, (ed. Martin Summerfield), American Institute of Astronautics and Aeronautics, Inc., 1988.

(41) Touchton, R.A., et.al., "Reactor Emergency Action Level Monitor Expert-System Prototype," EPRI NP-5719, 1988.

(42) Wolfgram, D.D., T.J. Dear, and C.S. Galbraith, Expert Systems for the Technical Professional. New York: John Wiley & Sons, Inc., 1987.

Neural Networks and Their Potential Application in Nuclear Power Plants

ROBERT E. UHRIG
Department of Nuclear Engineering
University of Tennessee
Knoxville, Tennessee, USA

and

Instrumentation and Control Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831, USA

ABSTRACT

A neural network is a data processing system consisting of a number of simple, highly interconnected processing elements in an architecture inspired by the structure of the cerebral cortex portion of the brain. Hence, neural networks are often capable of doing things which humans or animals do well but which conventional computers often do poorly. Neural networks have emerged in the past few years as an area of unusual opportunity for research, development and application to a variety of real world problems. Indeed, neural networks exhibit characteristics and capabilities not provided by any other technology. Examples include reading Japanese Kanji characters and human handwriting, reading a typewritten manuscript aloud, compensating for alignment errors in robots, interpreting very "noise" signals (e.g. electroencephalograms), modeling complex systems that cannot be modelled mathematically, and predicting whether proposed loans will be good or fail. This paper presents a brief tutorial on neural networks and describes research on the potential applications to nuclear power plants.

INTRODUCTION TO NEURAL NETWORKS

Neurons. The human brain is a complex computing device capable of thinking, remembering, and solving problems. There have been a number of attempts to emulate the brain functions with a computer model, and generally these have involved the simulation of a network of neurons, commonly called neural networks. The brain contains approximately 100 billion neurons that are densely interconnected with one thousand to ten thousand connections per neuron.

A neuron is the fundamental cellular unit of the brain's nervous system. It is a simple processing unit that receives and combines signals from other neurons through input paths called dendrites. (The basic components of a neuron are shown in Figures 1 and their schematic equivalents in Figure 2.) If the combined signal from all the dendrites is strong enough, the neuron "fires", producing an output signal along a path called the axon. The axon splits up and connects to hundreds or thousands of dendrites (input paths) of other neurons through synapses (junctions containing a neurotransmitter fluid that controls the flow of signals) located in the dendrites. Transmission of the signals across the synapses are electro-chemical in nature, and the magnitudes of the signals depend upon the synaptic strengths of the synaptic junctions. The strength or conductance (the inverse of resistance) of a synaptic junction is modified as the brain "learns". In other words, the synapses are the basic "memory units" of the brain.

Computer Simulation. The computer simulation of this brain function usually takes the form of artificial neural systems which consists of many artificial neurons, usually called processing elements or neurides. These processing elements are analogous to the neuron in that they have many inputs (dendrites) and combine (sum up) the values of the inputs. This sum is then subjected to a nonlinear filter usually called a transfer function, which is usually a threshold function or a bias in which output signals are generated only if the output exceeds the threshold value. Alternately, the output can be a continuous function (linear or nonlinear) of the combined input. Sometimes the outputs are "competitive" in which only one processing element has an output, or the processing elements may operate asynchronously in which outputs occur only when inputs occur simultaneously.

The output of a processing element (axon) branches out and becomes the input to many other processing elements. These signals pass through connection weights (synaptic junctions) that correspond to the synaptic strength of the neural connections. The input signals to a processing element are modified by the connection weights prior to being summed by the processing element. There is an analogy between a processing element and an operational amplifier in an analog computer in which many inputs are summed. The potentiometer settings on the amplifier inputs correspond to the connection weights and the output of the operational amplifier goes through some sort of nonlinear function generator.

NEURAL NETWORKS

Neural Networks. A neural network consists of many processing elements joined together to form an appropriate network with weighting functions for each input. These processing elements are usually organized into a sequence of layers with

full or random connections between layers. Typically, there are three or more layers: an input layer where data are presented to the network through an input buffer, an output layer with a buffer that holds the output response to a given input, and one or more intermediate or "hidden" layers. A typical neural network arrangement is shown in Figure 3.

The operation of an artificial neural network involves two processes: learning and recall. Learning is the process of adapting the connection weights in response to stimuli presented at the input buffer. The network "learns" in accordance with a learning rule which governs how the connection weights are adjusted in response to a learning example applied at the input buffers. Recall is the process of accepting an input and producing a response determined by the learning of the network.

Learning. There are several different kinds of learning commonly used with neural networks. Perhaps the most common is the so-called supervised learning in which a stimulus is presented at the input buffer of the network and the output from the output buffer is sent to a system that compares it with a desired output and then uses a corrective or learning algorithm to convert the difference (error signal) into an adjustment of the weighting coefficients (connection weights) that control the inputs to the various processing elements. A typical supervised learning system is shown in Figure 4. In a typical situation, the initial weighting functions are set randomly and then subjected to incremental changes determined by the learning algorithm. When an input is again applied to the input buffer, it produces an output which again is compared with the desired output to produce a second error signal. This iterative process continues until the output of the artificial neural network is substantially equal to the desired output. At that point, the network is said to be "trained". Through the various learning algorithms, the network gradually configured itself to achieve the desired input-output relationship or "mapping".

There are several other kinds of learning that are commonly used. For instance, in unsupervised learning, only the input stimuli are applied to the input buffers of the network. The network then organizes itself internally so that each hidden processing element responds strongly to a different set of input stimuli. These sets of input stimuli represent clusters in the input space (which often represent distinct real-world concepts). There is also "random learning" in which random incremental changes are introduced into the weighting functions, and then either retained or dropped depending upon whether the output is improved or not (based on whatever criteria the user wants to apply). A fourth type of learning is "graded" learning in which the output is graded on some numerical scale or perhaps simply classified as "good" or "bad" and then the connection weights are adjusted in accordance with the grade assigned to the output.

The common learning algorithms are: 1) Hebbian learning where a connection weight on an input path to a processing element is incremented if both the input is high (large) and the desired output is high. This is analogous to the biological process in which a neural pathway is strengthened each time it is used, 2) Delta-rule learning in which the error signal (difference between the desired output response and the actual output response) is minimized using a least-squares process, and 3) competitive learning in which the processing elements compete among each other and only the one that yields the strongest response to a given input modifies itself to become more like the input. In all cases, the final values of the weighting functions constitutes the "memory" of the neural network.

In the recall process, a neural network accepts a signal presented at the input buffer and then produces a response at the output buffer that has been determined

by the "training" of the network. The simplest form of recall occurs when there are no feedback connections from one layer to another or within a layer (i.e., the signals flow from the input buffer to the output buffer in what is called a "feed forward" manner). In this type of network the response is produced in one cycle of the computer. When the neural networks have feedback connections, the signal reverberates around the network, across the layers or within layers, until some convergence criteria is met and a steady-state signal is presented to the output buffers.

Characteristics of Neural Networks. The characteristics that make neural network systems different from traditional computing and artificial intelligence are 1) learning by example 2) distributed associative memory 3) fault tolerance and 4) pattern recognition.

The memory of a neural network is both distributive and associative. Distributed means that the storage of a unit of knowledge is distributed across all memory units (connection weights) in the network. This knowledge shares these memory units with all other items of knowledge stored in the network. Associative means that when the trained network is presented with a partial input, the network will choose the closest match to that input in its memory and generate an output that corresponds to the full output.

Traditional computer systems are rendered useless by any damage to its memory. However, neural-computing systems are fault tolerant in that if some processing elements are destroyed or disabled or have their connections altered incorrectly, the behavior of the network is changed only slightly. As more processing elements are destroyed, performance degrades gradually, i.e., the network performance suffers but the system does not fail catastrophically. This is because the information is not contained in any single memory unit, but rather is distributed among all the connection weights of the network. Such arrangements are well-suited for systems where failure may be unacceptable or introduce difficult problems (e.g., in nuclear power plants, missile guidance, and space probes.

Pattern recognition is the ability to match large amounts of input information simultaneously and generate a categorical or generalized output. It requires that the network provide a reasonable response to noisy or incomplete inputs. Experience shows that neural networks are very good pattern recognizers which also have the ability to learn and build unique structures for a particular problem.

NEURAL COMPUTING AND APPLICATIONS

Neural-computing networks consists of interconnected units that act on data instantly in a massively parallel manner. This provides an approach that is closer to human perception and recognition than conventional computers and can produce reasonable results with noisy or incomplete inputs. Neural computing is at an early stage of development. The results to date have been impressive, and they appear to complement expert systems. Future applications appear unlimited, but much development work remains to be done. A few of the recent applications of neural networks are given below to illustrate the wide spectrum of applications to which neural networks have been applied.

1. Complex system modeling. A system with multiple inputs and outputs can be modeled using a neural network by applying the system inputs to the network and using the system outputs as the desired outputs of the neural network. After an appropriate number of iterative learning cycles the neural network then constitutes a nonstructured non-algorithmic model of the process involved. Such modeling can be used on physical systems, business and financial systems, or

social systems. Current applications include the use of a neural network to determine whether loan applications should be approved using the previous five years experience of that bank as the input training data.

2. Image (data) compression involves the transforming of image data to a different representation that requires less memory. Then the image must be reconstructed from this new representation in such a way that there is an imperceptible difference from the original. Compression ratios of several hundred to one have been achieved in some cases.

3. Character recognition, a special case of pattern recognition, is the process of visually interpreting and classifying symbols. Neural networks were the first systems to efficiently read Japanese Kanji characters. This it effectively broke the input barrier for computers used in Japan.

4. Handwriting recognition involves a neural computing system that accepts handwriting on a digitized pad as a computer input and is trained by interpreting a set of handwriting types. The system can then interpret a type of handwriting it has never seen before and can make a "best guess" when confronted with a confusing character. Accuracy improves when the training is on the type of writing being read (e.g., on one individual's handwriting). A recent advance in Japan uses a process that simulates the way visual information feeds forward in the brain and has the advantage that it can recognize patterns regardless of orientation or distortion.

5. Target classification. Neural networks have been used to classify sonar targets by distinguishing between large metal cylinders and rocks of a similar size. The neural networks integrates 60 spectral energy values produced from 60 frequency bands. Its performance was comparable to the best trained human operators on the same data and significantly better than normal operators or other computer-based classifiers.

6. Noise filtering. Neural networks are able to filter noisy data and preserve a greater depth of structure and detail than any of the traditional filters while still removing the noise. Applications include removal of background noise from voice communications and separation of the fetal heart beat from a mother's heart beat.

7. Servo-control systems. Complex mechanical servo-systems, such as those used in robots, must compensate for physical variations in the system introduced by misalignments in the axes, or deviation in members due to bending and stretching induced by loads. These quantities are extremely difficult to describe analytically. A neural network can be trained to predict and respond to these errors in the final position of a robot member. This information is then combined with the desired position to provide an adaptive position correction and improve the accuracy of the member's position.

8. Text-to-speech conversion. In this application the printed symbols or letters in a text were converted into the spoken language using a neural network that taught itself to translate written text into speech in the same way that a human child learns to read. The printed transcript is broken down into the fundamental components of speech called "phonemes" which became the desired output of the neural network when the input was the corresponding text. After training, the phonemes become the input to a voice synthesizer which provides the verbal output.

APPLICATIONS TO NUCLEAR POWER PLANTS

When a complex system plant is operating safely, the outputs of hundreds, or even thousands, of sensors or control room instruments form a pattern (or unique set) of readings that represent a "safe" state of the plant. When a disturbance occurs, the sensor outputs or instrument readings form a different pattern that represents a different state of the plant. This latter state may be safe or unsafe, depending upon the nature of the disturbance. The fact that the pattern of sensor outputs or instrument readings is different for different conditions is sufficient to provide a basis for identifying the state of the plant at any given time. To implement a diagnostic tool based on this principle, that is useful in the operation of complex systems, requires a rapid (real-time), efficient method of "pattern recognition." Neural networks offer such a method.

Useful Features of Neural Networks. Neural networks may be designed so as to classify an input pattern as one of several predefined types of faults or transients (e.g., the various fault or transient states of a power plant) or to create, as needed, categories or classes of system states which can be interpreted by a human operator. Neural networks have demonstrated high performance even when presented with noisy, sparse and incomplete data.

A second desirable feature of neural networks is their ability to respond in real-time to the changing system state descriptions provided by continuous sensor input. For complex systems involving many sensors and possible fault types (such as nuclear power plants), real-time response is a difficult challenge to both human operators and expert systems. However, once a neural network has been trained to recognize the various conditions or states of a complex system, it only takes one cycle to detect a specific condition or state. Because neural networks can be trained to recognize the patterns of different sensor outputs or instrument readings that give rise to different system states or faults, they are ideally suited for real-time diagnostics.

Neural networks have the ability to recognize patterns, even when the information comprising these patterns is noisy or incomplete. Unlike most computer programs, neural network implementations in hardware are very fault tolerant; i.e. neural network systems can operate even when some individual nodes in the network are damaged. The reduction in system performance is about proportional to the amount of the network that is damaged. Thus, systems of artificial neural networks have high promise for use in environments in which robust, fault-tolerant pattern recognition is necessary in a real-time mode, and in which the incoming data may be distorted or noisy. This makes artificial neural networks ideally suited as a candidate for fault monitoring and diagnosis, control, and risk evaluation in complex systems, such as nuclear power plants.

NEURAL NETWORK PROJECTS AT THE UNIVERSITY OF TENNESSEE

In October 1988, a three-year Department of Energy contract entitled "Enhancing the Operation of Nuclear Power Plants through the Use of Artificial Intelligence" was initiated at the University of Tennessee at a funding level of about $250,000 per year. This program calls for a number of investigations of how both expert systems and neural networks can be applied (both on-line and off-line) to enhance the overall operation of nuclear power plants. Most of these projects involve the use of computers to carry out the work and to simulate a nuclear power plant or certain plant systems in order to demonstrate the feasibility of the process being investigated. The investigations currently in progress under this program include:

1. Multi-sensor fusion. This project has as its goal the integration of diverse types of signals (temperature, pressure, neutron flux, etc.) at many locations (in the core, the primary coolant loop, the steam generators, etc.) in order to provide a reasonable representation of the processes involved in a nuclear power plant. The method involves the adaption of neural network technology from "scene representation." This project is being carried out at the University of Tennessee Space Institute in Tullahoma, Tennessee under the direction of Dr. Alianna J. Maren.

2. Signal validation. This project is investigating the feasibility of using neural networks to validate the signals coming from many sensors of the same or similar type located in a subsystem of a plant. The acceleration of the backpropagation network training and minimization of signal prediction error are being studied.[1] This project is being carried out under the direction of Dr. Belle Upadhyaya in the Department of Nuclear Engineering at University of Tennessee, Knoxville.

3. Nuclear fuel management. This project is investigating the feasibility of using neural networks to significantly reduce the computation involved in establishing core reload configurations in nuclear power plants. A separate investigation involves the application of the optimization methods of the "traveling salesman problem" to optimizing the fuel element pattern for some specific parameter (e.g., economy, minimum leakage of neutrons from the core, minimum uranium consumption, lowest peaking factor, etc.). This project is being carried out by Dr. Laurence Miller in the Department of Nuclear Engineering at the University of Tennessee, Knoxville.

4. Modeling and Diagnostics in Nuclear Power Plants. This project has as its goal the development of diagnostic methods using neural networks to detect faults and transients in nuclear power plants. In some cases, this will involve the development of non-structured, non-algorithmic models of process or major components using neural network techniques. In other cases, this model will be part of the knowledge base of an expert system. This work is being carried out under the direction of Dr. Robert E. Uhrig in the Department of Nuclear Engineering at the University of Tennessee.

5. Prediction of Energy Needs of the United States using Neural Networks. This project has as its goal the development of multiple neural networks to predict the future energy needs in the United States. The underlying goal of this project is not just the prediction of energy needs, but rather, the development of a methodology using multiple neural networks which may also have application to the diagnostic processes. This work is being carried out under the direction of Dr. Robert E. Uhrig in the Department of Nuclear Engineering at the University of Tennessee.

6. Neural Network Algorithms that "Learn" in a Single Cycle. A new approach to learning in neural networks has been developed that allows the network to learn to recognize a pattern in a single cycle. It uses competitive learning in the "hidden" layer and can learn as many patterns as there are processing elements in this layer. Both uni-directional and bi-directional versions of this algorithm have been demonstrated. This work is being carried out under the direction of Dr. Robert E. Uhrig in the Department of Nuclear Engineering at the University of Tennessee, Knoxville.

Some of these projects involve only simulation on computers to demonstrate proof of principle. Many of them can benefit significantly by utility involvement to demonstrate their usefulness in nuclear power plants. Some of them will require extensive testing in a training or engineering simulator of a nuclear power plant

prior to implementation in nuclear power plants. For instance, the use of neural networks to identify abnormal conditions or transients in nuclear power plants would require extensive use of a sophisticated nuclear power plant simulator, preferably a full-fidelity simulator of a particular nuclear power plant.

IDENTIFICATION OF SYSTEM TRANSIENTS

Let us look at a typical application of neural networks, the identification of system transients in a nuclear power plant. The goal is the demonstration of the feasibility of using a neural network to diagnose different fault conditions or transients in a nuclear power plant. The initial task was the simulation of transients of a steam generator of a pressurized water reactor. For demonstration purposes, a set of data from a simulation of an isolated U-tube steam generator (UTSG) is used as the training data for the neural network.[2,3]

For this demonstration, only six step-change perturbations were introduced into the system:

1. Positive and negative step changes in primary inlet temp.,
2. Positive and negative step changes in feedwater temp., and
3. Positive and negative step changes in the valve coefficient (percentage of the full-open position).

Four variables (i.e., sensor outputs or instrument readings) were chosen as the system responses to represent the system behavior for each perturbation. They were: the primary inlet temperature, the primary outlet temperature, the downcomer water level, and the steam pressure. The time-records of these four quantities show that the patterns are quite different for each of the six perturbations listed above. Since the back-propagation network can be trained to distinguish between the different patterns, it can distinguish between the different kinds of steam generator perturbations.

The time-records of these six perturbations (at the 10°F and 10% perturbation levels) constitute the input pattern for training the back-propagation neural network. Ten samples were taken at ten second intervals for each of the four variables from the response curves (a total of 40 values) for each of the six simulations, digitized, and used as inputs to the neural network. The desired outputs for training of the network were defined by a 3-bit binary quantity, representing the output states of the six perturbations.

In this study, a three layer back-propagation network was set up with 40 processing elements (PEs) in the input layer and three PEs in output layer to match the dimensions of the training input and output vectors. The middle layer contained 12 PEs because this size represented a reasonable compromise between ease of training and precision for the number of inputs and outputs.

The digitized time-record for the perturbations of $\pm$10°F in primary inlet temperature, $\pm$10°F in feedwater water temperature, and $\pm$10% in valve opening coefficient were used as six sets of training data for the network. The data were normalized before putting them into the neural network to satisfy the requirement of the input form of the neural network used. After 500 data training cycles using the back-propagation learning algorithm (which may require anywhere from a few seconds to a few hours, depending on the computer hardware and software used), the network readily distinguished between the six different perturbations. Random fluctuations were introduced into the trained network, and the recall results

showed that the network could still identify each of the UTSG transients correctly, even when the amplitude of the noise was equal to 90% of the amplitude of the signals.

Acknowledgement. Some of the work described here was supported by the U. S. Department of Energy under contract # DOE DEFG-88ER12824, entitled "Enhancing the Operation of Nuclear Power Plants Through the Use of Artificial Intelligence." This support is gratefully acknowledged.

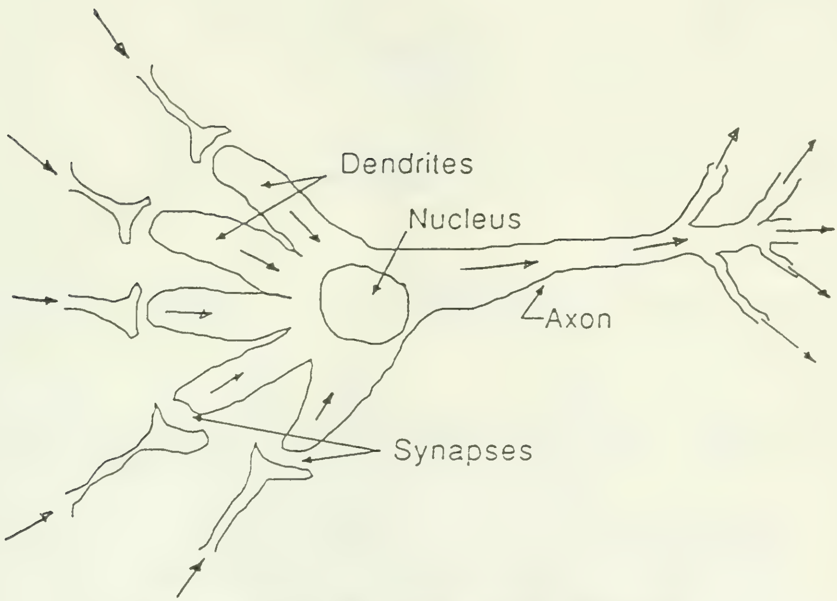


FIGURE 1. SKETCH OF A NEURON SHOWING COMPONENTS

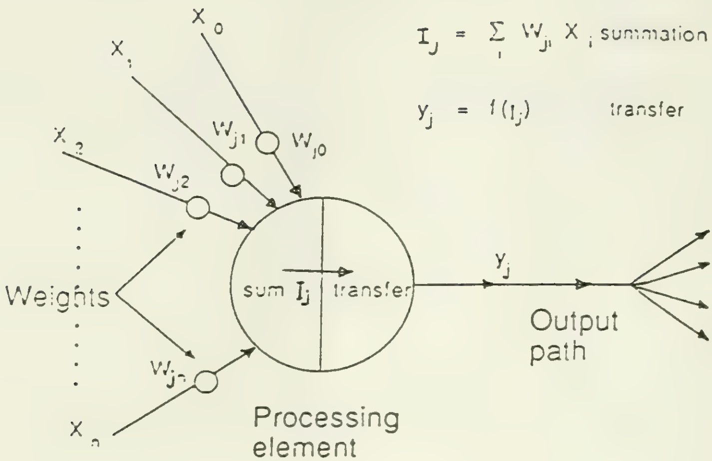


FIGURE 2. SCHEMATIC REPRESENTATION OF A NEURON

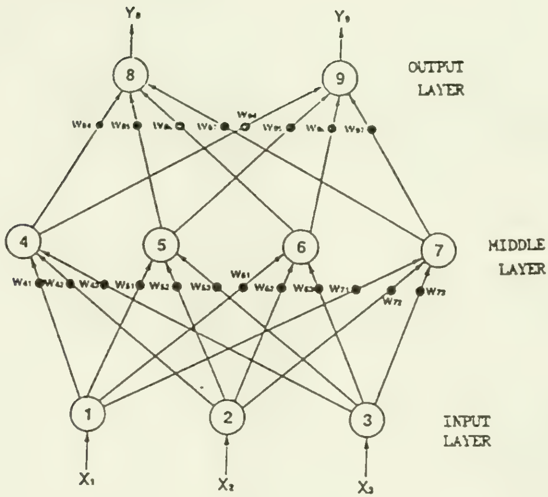


FIGURE 3. A SIMPLE 3-LAYER NEURAL NETWORK

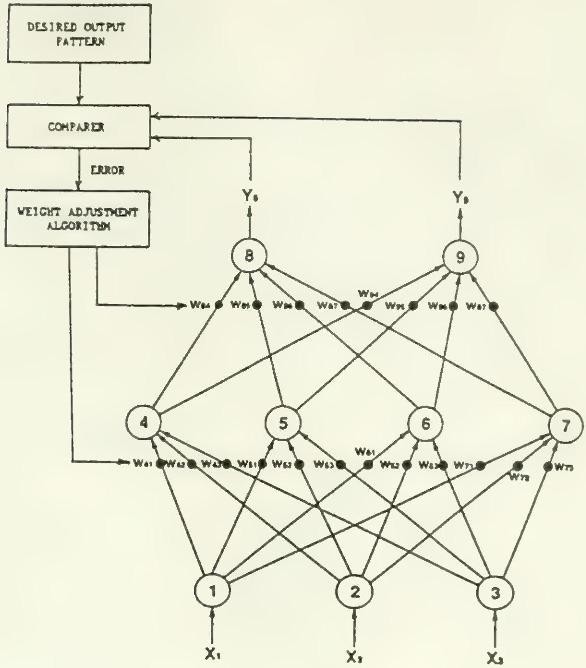


FIGURE 4. A TYPICAL SUPERVISED LEARNING SYSTEM

References

1. B. R. Upadhyaya, E. Eryurek, G. Mathai, "Application of Neural Computing Paradigms for Signal Validation," Seventh Power Plant Dynamics, Control and Testing symposium, Knoxville, TN, May 1989.

2. R. E. Uhrig, Z. Guo, "Use of Neural Networks in Diagnoses of Transient Operating Conditions Using a Back-Propagation Neural Network," Proceedings of SPIE-the International Society for Optical Engineering VII, "Application of Artificial Intelligence," Orlando, FL, March 28-30, 1989.

3. T. W. Kerlin, E. M. Katz, "Pressurized Water Reactor Modeling for Long-Term Power system Dynamics," EPRI EL-3087, Volumes 1, 2, May 1983.

Index

ISBN 1-56032-102-4